Automatic Gain and Fader Control For Live Mixing

Enrique Perez_Gonzalez and Joshua Reiss

Center For Digital Music
Queen Mary University of London
Mile End Road, London, E1 4NS, UK
enrique.perez | josh.reiss@elec.qmul.ac.uk

ABSTRACT

A cross-adaptive mixing device has been developed for the purpose of optimizing the gain levels of a live audio mixture. The method aims to achieve optimal mixing levels by optimizing the ratios between the loudness of each individual input channel and the overall loudness contained in a stereo mix. In order to evaluate the amount of loudness of each channel in real-time, accumulative statistical measurements were performed. The system uses a cross-adaptive algorithm to map the loudness indicators to the channel gain values. The system has applications in automatic mixing of live music, live mixing of game audio, and studio recording post-production.

1. INTRODUCTION

In order to create a balanced audio mixture, careful scaling of input gains and level faders must be achieved. Several methods for automatically setting levels for speech have been proposed, notably [1-4]. On the other hand, only a few methods for automatically setting the levels for music have been proposed by [1, 5, 6]. In all three cases, the systems are based on measuring signal amplitude and adapting the mixture according to amplitude indicators. The use of perceptual attributes was suggested by [6], but was not implemented as such. Method [1] works by turning on and off microphones when their input level is greater than an adaptive threshold, while [5] attempts to achieve mixture balance by lowering high amplitude signals and increasing the level of low amplitude signals.

In this paper, the authors approach the problem by making use of cross-adaptive methods driven by a perceptual indicator. Cross-adaptive algorithms are based on inter-channel dependency processing. In other words, they are multiple input, multiple output algorithms, in which individual channel processing is dependent on the interaction with other channels. The proposed system attempts to handle the task of weighting the gain between channels by using accumulative loudness measures. We assume that a mixture in which loudness per channel tends to the overall average loudness is a well-balanced mixture with optimal inter-channel intelligibility. By doing this, each channel has an equal chance of masking other channels, thus optimising the likelihood of each channel being heard. To achieve optimisation, the system adapts its gain according to the relationship of loudness indicators between channels and the overall average loudness of the mix. In this paper, the authors will present the theory and implementation behind such a system, and results demonstrating the effectiveness of our technique.

2. SCALING THE AUDIO SIGNALS

Given that the electronic components of a mixer have amplitude limits before distortion, the input signals may be said to have a maximum value of 1. Normalizing all inputs to have maximum peak value of 1 would optimize the dynamic range of the system while giving a common reference for all inputs. For this reason a mechanism to prevent exceeding this limit should be implemented. Unfortunately, in a live system we do not know the maximum level of the incoming signal. Therefore, adaptive gain compensation should be implemented.

Consider a sound mixture $X(t)$ is comprised of M channels and each individual channel input $f_m(t)$ has a scaling factor $k_m(t)$, also known as a channel fader level. We can say that the total mixture in the time domain is given by the summation from 1 to M of $k_m(t)\,f_m(t)$, where $k_m(t)$ and $f_m(t)$ can take a maximum value of 1.

In order to avoid distortion on the inputs, $f_m(t)$ must be scaled continuously by an input gain factor $g_m(t)$ according to the following equation,

$$X(t) = \sum_{m=1}^{M} k_m(t) f_m(t) g_m(t) \tag{1}$$

where $g_m(t)$ is the adaptive rescaling factor of $f_m(t)$ in order to avoid distortion. $g_m(t)$ is equal to 1 for t_0 and $g_m(t) = g_m(t-1)-r$, where r is the amount of decrement applied every time $f_m(t)$ is greater than 1. This method was implemented on a recallable head amplifier digital to analogue converter. The implementation is depicted in Figure 1.

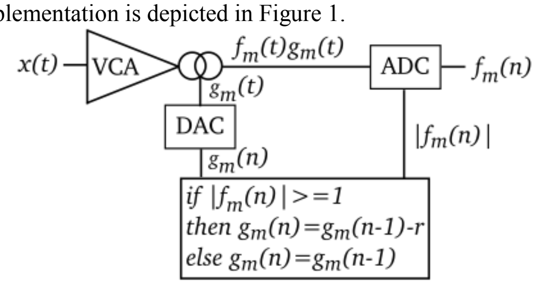

Figure 1: Adaptive Gain Signal Acquisition block diagram.

The implementation in Figure 1 consists of a hybrid circuit in which the system is capable of imputing an analog signal while outputting a digital one. Based on such an implementation we can use the overflow flag of the ADC, $|f_m(n)|$, for determining if there is distortion on the ADC due to excessive head amplifier gain. Since the head amplifier is a Voltage Controlled

Amplifier, VCA, we can control the amount of signal coming into the ADC by driving it with $g_m(t)$ as its adaptive rescaling factor. Given that the ADC is reporting the overflow signal in the form of a digital bit, and a bit is equal to 6dBfs where the bit has two states each compromising a 3dBfs, then the optimal value for r is -3dBfs. This design will ensure correct digitization of the signal while continually adapting the gain in the case of signal distortion. Most importantly, this will give the system a normalized set of input signals $f_m(n)$, for which the reference limits are the same.

3. LOUDNESS ESTIMATION

In the same form that [1] stated that the success of their method was dependent on the implementation of an amplitude envelope feature, our implementation is subject to the overall performance of the loudness feature. Loudness is a perceptual attribute of sound and therefore requires a psycho-acoustic model. Our psycho-acoustic model implementation has been depicted in Figure 2.

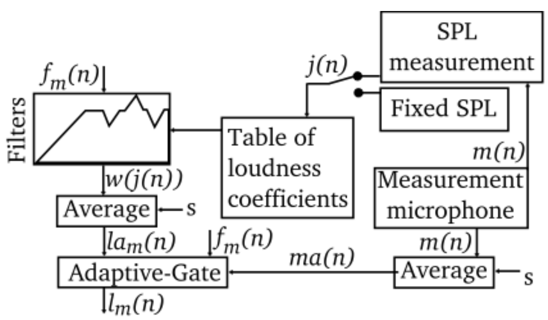

Figure 2: Loudness feature block diagram.

The loudness feature implementation for this paper utilizes the ISO 226 standard loudness curves [7]. The curves are used to weight the amplitude of the sampled input signal $f_m(n)$. A loudness weighting curve, $w(j(n))$, was stored for every $10dB_{SPL}$ increment, within a range of 10 dB_{SPL} to $130dB_{SPL}$, where $j(n)$ states the increment range of the weighting curve. Four biquadratic filters in series were used to approximate the loudness curves. The coefficients for each biquadratic filter, corresponding to each loudness curve, were stored in a lookup table. The lookup table outputs the appropriate coefficients according to the reading of a sound pressure level meter device, or can be manually fixed to a desired value for non-live applications. The input loudness per channel is calculated and reported to the system by using (2).

$$la_m(n) = \sum_{n=1}^{s} f_m(n) w(j(n)) / s \tag{2}$$

Where $j(n)$ corresponds to the measured sound pressure level, and s represents a given number of samples for calculating the bipolar mean amplitude loudness, $la_m(n)$. Our current implementation has average buffer size of $s=200$ samples. Although the calculation is performed per channel, the value taken by $j(n)$ is the same for all channels. Thus all channels are weighted with the same loudness curve.

3.1. Adaptive gating

In practice, a noisy input distorts the loudness measurement. For this reason a gate with an adaptive threshold was implemented. Considered the inputs being live microphone instruments on a stage. There is a usable distance where the microphone performs well. If the performer is too far away from the microphone, the background noise captured by the microphone increases, making the microphone signal to noise ratio too low and unsuitable for reproduction. In [1], a method of installing a measurement microphone outside the usable distance of the reproduction microphones was proposed as a measurement for noise. This measurement of noise $m(n)$ may be used as the threshold for the adaptive gating. In other words the system will only let a signal through if $f_m(n)$ is $\geq m(n)$. This gated signal represents a cleaner, more representative measurement of the loudness of $f_m(n)$, and will be noted as $l_m(n)$. When gated, $l_m(n)$ is in a state of no signal, which is different from a state of silence. This is important given that a correct loudness measurement can now be achieved which is not biased by silence.

3.2. Accumulating the loudness

Once we have a clean measurement of $l_m(n)$, we can proceed to analysis. The implementation proposed in this paper is the use of accumulative histograms of loudness. These give a representation of the loudness mass probability function, therefore representing its probabilistic behavior from the start of the measurement up to the point in time of the current measurement. Since the system is to be used in real time, for computing a histogram we must consider the possible minimum and maximum ranges of the loudness signal. This is done to ensure the maximum number that can be held by the histogram function is equal to the maximum value taken by the function $l_m(n)$. This is analogous to ensuring the level of a preamplifier is such that it avoids clipping. Actually the system is capable of having a maximum amplitude input of one, the actual measurement of such a signal can generate a loudness measure greater than or less than one. For this reason, a self-adjusting scaling mechanism was implemented to ensure that the values of $l_m(n)$ were within the range of 0 to 1. The normalization algorithm scans for a probability higher than zero on its highest bin, Bin_{max}. In case this is true for any of the channels, then the rescaling gain of all channels, $r(n)$, should be decreased by a factor d. The process should be repeated recursively until the highest bin in the histogram is equal to zero for all channels. All channels must use the same gain scalar in order to have a common reference, so the gain of the channel with highest input level is used. Such a system is depicted in Figure 3.

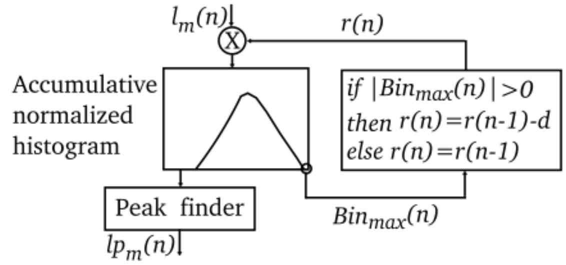

Figure 3: Histogram adaptive rescaling.

The current implementation has a decrement value $d=0.5$ and a rescaling start gain of $r(0)=100$. These values have been determined experimentally, although the system is robust to parameter changes. Once the accumulative histograms have been correctly rescaled and gated, we can proceed to calculate the highest peak for each channel and use it as the most probable loudness state, $lp_m(n)$.

4. CROSS ADAPTIVE FUNCTION

Our aim is to adapt $k_m(n)$, the channel fader level, continuously for each channel such that we achieve a common average loudness between channels. For this we use the model depicted in Figure 4.

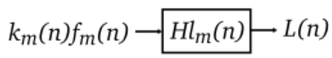

Figure 4: Loudness feature system diagram.

$L(n)$ is the result of averaging all $lp_m(n)$ of the system and $Hl_m(n)$ is the transfer function of the combined block diagram represented in Figure 2 and 3. $f_m(n)$ represents the channel input and $k_m(n)$ represents the gain weighting factor of each channel, which is modified to achieve the target $L(n)$. Given this model we can derive the following equation for determining $k_m(n)$,

$$k_m(n) = \frac{L(n)}{Hl_m(n) f_m(n)} \tag{3}$$

where $k_m(n)$ represents the fader gain per channel in order to achieve $L(n)$. The function of the combined block diagram depicted in Figures 2 and 3, is given by $Hl_m(n)=lp_m(n)/f_m(n)$. So $Hl_m(n)f_m(n)=lp_m(n)$, where $lp_m(n)$ corresponds to the most probable loudness state per channel. This proposed model has the advantage that it is not dependent on the feature used, thus leaving room for future study of better features without the need for a major re-implementation of the system.

4.1. Determining the headroom of the system

In a real application, $k_m(n)$, has physical limitations. In most cases $k_m(n)$ will be represented by a physical fader with range limits. The system must ensure that the values of $k_m(n)$ are within range. The proposed solution is to scale the measured $f_m(n)$ before it is measured. This scaling is proportional to the available headroom that the system will have between $L(n)$ and the maximum value that can be taken by $k_m(n)$. For example, scaling 0.5 $f_m(n)$ will give a 6dB headroom to the mix with respect to $L(n)$. This rescaling is currently user selectable and must be selected according to the type of dynamics of the music being mixed. If a channel requires a compensation which forces $k_m(n)$ to go out of range, it is clamped to its highest possible value. Such a clamping action should indicate to the user the need for compressing this particular signal or re-selecting the microphone position. In order to apply this headroom scaling factor, equation (3) must be updated in the following manner:

$$k_m(n) = \frac{L(n)}{Hl_m(n) f_m(n) hr} \tag{4}$$

where hr correspond to the available headroom fader level.

4.2. Keeping overall stability of the system

Given that the electronic components of the output of a mixer have amplitude limits before distortion. So the overall mixture and individual signals would be said to have a maximum amplitude limit of one. If we design a system in which this is the case regardless of the values taken by $k_m(n)$, then maximum gain before feedback will be maintained [8]. For implementing such a design we must continuously normalize the gain values to add up to unity gain. Such a method, which we refer as cross normalization, has been suggested by [1]. It is important to mention that suitable interpolation methods are required in order to implement this method in real time applications. This will ensure continuous gain level changes with no audible undesired artifacts. The equation for achieving such normalization is given next,

$$ks_m(n) = \frac{k_m(n)}{\sum_{m=1}^{M} k_m(n)} \tag{5}$$

where $ks_m(n)$ is the normalized version of $k_m(n)$. Thus the summation of all $ks_m(n)$ from 1 to M is equal to one. Therefore the final mixture is given by the following equation,

$$X(n) = \sum_{m=1}^{M} ks_m(n) f_m(n) \tag{6}$$

where $f_m(n)$ input gain has been previously established by $g_m(n)$ and $ks_m(n)$ has been obtained by targeting a common loudness average $L(n)$ between all channels in the mixture, while doing a cross normalization. An overall system diagram is depicted next:

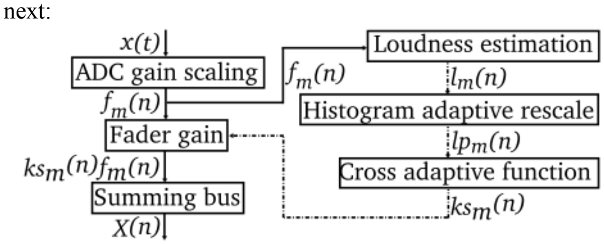

Figure 5: Overall system diagram. Solid line audio path, dotted line data control path.

5. RESULTS

A high gain musical test signal, $f_m(t)$, shown in Figure 6B was used as an input for a simulated block diagram depicted on Figure 1. The self adjusting gain factor, $g_m(n)$, was updated every time $f_m(n)$ had an amplitude higher than one. This is depicted in Figure 6A. It can also be seen that the algorithm has a fast convergence. The resulting $f_m(n)$, which is within the maximum range of +/- one, is depicted in Figure 6C.

Using a musical test signal $f_m(n)$ as an input, it was noted that if no rescaling and no adaptive gating was used to optimize the loudness mass probability function, the resulting $lp_m(n)$ was always 0, as shown in Figure 7A. This is because there is a large amount of low-level noise and zero crossing which bias the loudness measurement. A second test with rescaling and no adaptive gating was done, as shown in Figure 7B. It can be see that although a gaussian shape corresponding to the actual loudness we desired to measure can be de seen, unfortunately there is still a large number of occurrences in the lowest bin of the histogram causing an erroneous null measurement. A third test consisting of adaptive gating and no rescaling was also performed, Figure 7C. Note how the amount of zero-bin occurrences has been dramatically reduced. Finally, a test consisting of both, rescaling and adaptive gating was performed, Figure 7D. It can be seen that the algorithm has been able to correctly identify $lp_m(n)$. This means that that adaptive rescaling and gating must be performed in order to achieve a clean $lp_m(n)$ loudness measurement.

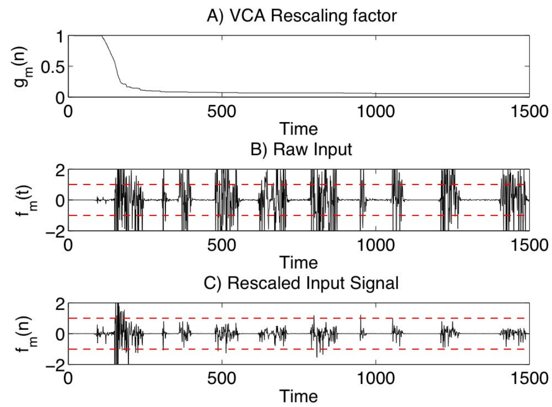

Figure 6: Simulation of automatic input gain normalization, [Time in units of 10ms].

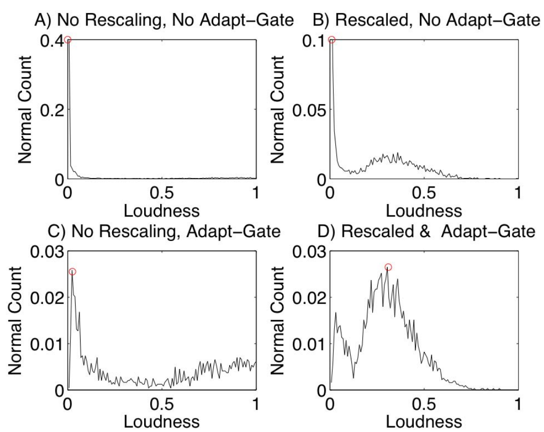

Figure 7: Accumulated loudness histograms $l_m(n)$. The circular marker denotes the resulting accumulated peak loudness value.

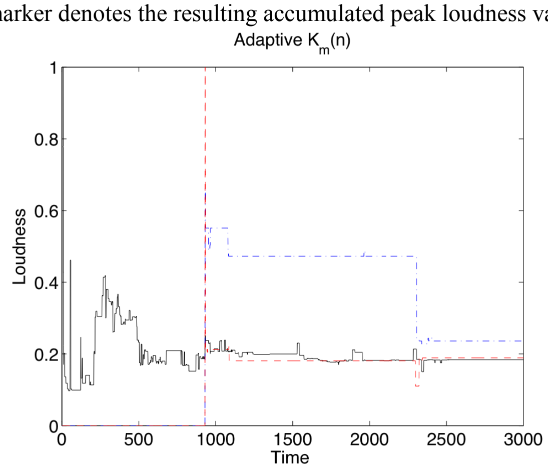

Figure 8: Cross-adaptive target loudness for a single music channel. $L(n)$ Average loudness (-), $lp_m(n)$ accumulated peak loudness (-.-). Post-$lp_m(n)$ accumulated peak loudness measured after Applyng the algorithm, (--).[Time in units of 10ms].

Finally, using a musical signal, a measurement of the convergence between the channel loudness $lp_m(n)$ and the overall mixture average loudness $L(n)$ was performed. This is depicted in Figure 8. The musical signal was a choir part of a song, which happens to start at time 10000ms. An extra measurement of the average peak loudness after the signal had been weighted with the calculated $k_m(n)$ was also performed. The resulting $lp_m(n)$, referred to as post-$lp_m(n)$, corresponds to the measured peak loudness of $f_m(n)k_m(n)$ given that, $k_m(n)$, has been calculated using equation (4). A user assigned headroom target, $hr=0.5$ was used. It can be seen that on average, post-$lp_m(n)$ is able to follow the magnitude of $L(n)$. It can also be seen that the accumulative characteristics of the measurement make it robust to noisy changes, although this could also prove troublesome in some cases where a sudden change in $k_m(n)$ is needed . For a video demonstration of the implemented system go to: http://www.elec.qmul.ac.uk/digitalmusic/automaticmixing/autofadegain.mov

6. CONCLUSIONS

An implementation of a cross-adaptive effect has been developed for the purpose of automatically optimizing the gain levels of an audio mixture. A mixing model in which the accumulated loudness per channel has equal chance has been proposed. Early performance tests performed on the system indicate that the method has potential applications in real-time automatic mixing. It is thought that the use of better psycho-acoustic models for loudness, such as [9], could yield better results. Thus the use of a better loudness feature should be explored. Trials on live performances as opposed to live recordings are yet to be performed. More system profiling is required in order to examine its subjective performance and address questions such as whether people like equal loudness mixes and how does the system compare to a human mixer.

7. REFERENCES

[1] D. Dugan, "Automatic Microphone Mixing," in *AES*, 1975.

[2] D. Dugan, "Application of Automatic Mixing Techniques to Audio Consoles," in *AES*, 1989.

[3] S. Julstrom, and T. Tichy, "Direction-Sensitive Gating: A New Approach to Automatic Mixing," *JAES,* vol. 32, pp. 490-506, Julay/August 1984.

[4] R. W. Peters, "Priority mixer control " in *USA Patent, 4.149.032*, 1978.

[5] E. Campbell, and Whittemore R. T., "Automatic microphone mixing apparatus " in *European Patent Office, US 4357492 (A)* 1982.

[6] R. B. Dannenberg, "An Intelligent Multi-Track Audio Editor," in *ICMC*, 2007, pp. 89 - 94.

[7] ISO, "Acoustics — Normal equal-loudness-level contours, ISO 226:2003(E) ", 2003.

[8] E. Perez_Gonzalez, and J. Reiss, "An automatic gain normalisation technique with applications to audio mixing," in *AES*, 2008.

[9] E. Skovenborg, and Lund T., "Loudness Descriptors to Characterize Programs and Music Tracks," in *AES*, 2008