Audio Engineering Society

Convention Paper 8961

Presented at the 135th Convention
2013 October 17–20 New York, USA

A knowledge-engineered autonomous mixing system

Brecht De Man[1], Joshua D. Reiss[1]

[1] *Centre for Digital Music, Queen Mary University of London, Mile End Road, London E1 4NS, United Kingdom*

Correspondence should be addressed to Brecht De Man (brecht.deman@eecs.qmul.ac.uk)

ABSTRACT

In this paper a knowledge-engineered mixing engine is introduced that uses semantic mixing rules and bases mixing decisions on instrument tags as well as elementary, low-level signal features. Mixing rules are derived from practical mixing engineering textbooks. The performance of the system is compared to existing automatic mixing tools as well as human engineers by means of a listening test, and future directions are established.

1. INTRODUCTION

Since the first automatic microphone mixer [1], many systems have been proposed to automate various mixing engineering tasks, such as balancing levels, panning signals between channels, dynamic range compression and equalisation [2–13]. However, these systems generally lack instrument-specific processing. Mixing decisions are based solely on the extracted, low-level features of the signals and no high-level semantic information, such as which instruments the incoming tracks accommodate or the genre of the song, is provided by the user or extracted by the system.

In this paper, we investigate a system that mixes raw audio tracks into a stereo track using balance, pan, compression and equalisation rules derived from practical audio engineering literature [14–21]. Additionally, equaliser and compression presets included with the digital audio workstation (DAW) Logic Pro 9 are added to the rule base.

These sources stress that mixing is highly non-linear [19] and unpredictable [21], and that there are no hard and fast rules to follow [19], "magic" settings [20] or even effective equaliser presets [21]. It should be noted that spectral and dynamic processing of

tracks does indeed depend very much on the characteristics of the input signal. This paper is by no means aiming to disprove that. Rather, it seeks to investigate to what extent semantic information about a project and its individual tracks, in combination with elementary low-level features, allows a system to make suitable mixing decisions.

To this end, we developed a framework that includes modules to read these rules, modules to measure elementary, low-level features of audio signals, and modules to carry out elementary mixing tasks (dynamic range compression, equalising, fading, panning) based on the rules.

Section 2 presents the system and a rule base derived from practical mixing engineering literature. We conduct a listening test to assess the performance of this system and compare it to another automatic mixing system (not knowledge-based and without track labels) as well as human mixing engineers, as described in Section 3. The results of this test are then discussed in Section 4. Section 5 covers the conclusions we drew from this experiment and outlines future directions.

2. SYSTEM

Figure 1 shows a block diagram of the proposed system.

The system's input consists of raw, multitrack audio (typically a mixture of mono and stereo tracks), and a text file specifying the instrument corresponding with every audio file (e.g. `Kick_D112.wav, kick drum`). Elementary features of every track are extracted at the measurement stage (see Section 2.2). For easy access within the system, the track number is automatically stored as an integer or integer array named after the instrument (e.g. if channel 1 is a kick drum: `kickdrum = 1`, if channels 3 through 5 are toms: `tom = [3, 4, 5]`). The different track indices are also stored in subgroup arrays (e.g. `drums_g = [1, 2, 3, 4, 5, 7, 12]`) to be able to access all guitars, vocals, ... at once. Then, rules are read from the rule base and, if applicable, applied to the respective input tracks. The rule specifies one out of five compressors: high pass filtering ('HPF'), dynamic range compression ('DRC'), equalisation ('EQ'), balance/level ('fader') and panning ('pan pot'). The order of the application of the

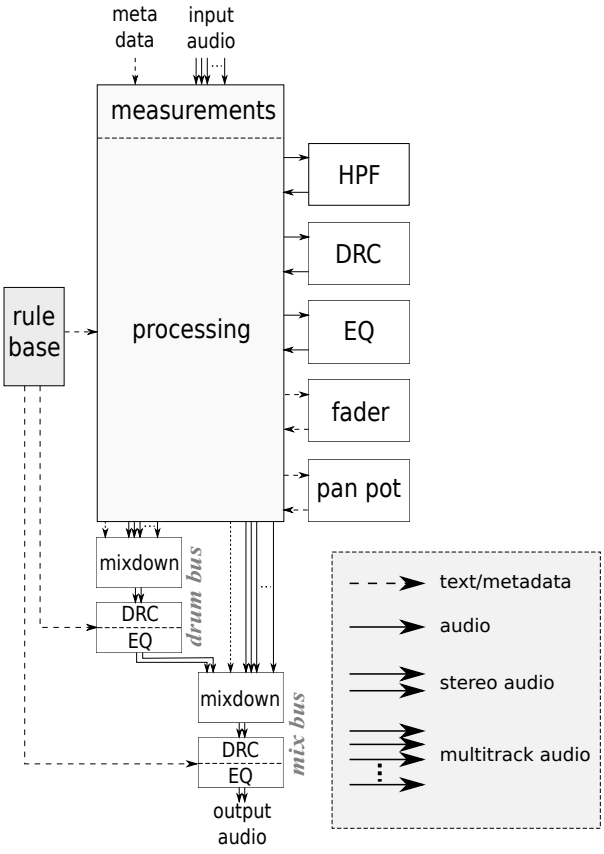

Fig. 1: Block diagram of the system. Solid arrows represent audio input or output; dashed arrows represent textual information such as instrument names and other metadata, and rules.

rules is determined by the chosen order of the processors, i.e. first the knowledge base is scanned for rules related to processor 1, then processor 2 and so on.

After processing the individual tracks, the drum instruments (members of subgroup `drums_g`) are mixed down using the respective fader and panning constants, and equalised and compressed if their are rules related to the drum bus. Eventually, the stereo drum bus is mixed down together with the remaining tracks, again with their respective fader and panning constants. The resulting mix is equalised and compressed if there are rules acting on the mix bus.

At this point, both the extracted features and the mixing parameters are constant over the whole of the audio track (in this experiment only short, four-bar audio fragments are used). In case longer audio tracks should be processed, it would be advisable to calculate these measures per song section (if sections are marked by the user or automatically), or have measures and settings that vary over time continuously.

2.1. Rule list

Each rule in the rule list consists of three parts:

- *tags*: comma-separated words denoting the source of the rule (sources can be included or excluded for comparison purposes), the instrument(s) it should be applied on (or 'generic'), the genre(s) it is applicable in (or 'all'), and the processor it concerns. Based on these tags, the inference engine determines if the rule should be applied, and on which track. The order and number of tags is not fixed.
- *rules*: The 'insert' processors (high-pass filter, compressor and equaliser) replace the audio of the track specified in the *tags* part with a processed version, based on the parameters specified in the *rules* part. This is done immediately upon reading the rule. The level and pan metadata manipulated by the rules, on the other hand, are not applied until the mixdown stage (see Section 2.3.5), after all the rules have been read. The rule can also contain other MATLAB code, like conditional statements, loops, or calculations. Audio and metadata corresponding to the processed track, as well as other tracks, can be accessed from within the rule.
- *comments*: These are printed in the console to show which rules have been applied, and to facilitate debugging.

An example of a rule is as follows:
tags: `authorX, kick drum, pop, rock, compressor`
rules: `ratio = 4.6; knee = 0; atime = 50; rtime = 1000; threshold = ch{track}.peak - 12.5;`
comments: `punchy kick drum compression`

In future work, conversion of the rules to a formal data model and use of the Audio Effects Ontology [22] will facilitate exchanging, editing and expanding the rule base, and enable use in description logic contexts.

2.2. Measurement modules

For every incoming track, the following quantities are measured and added to the track metadata: the number of channels (mono or stereo), RMS level (1a), peak level (1b), crest factor (1c) and loudness (following the definition from [26]).

$$L_{rms} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}|x(n)|^2} \tag{1a}$$

$$L_{peak} = \max(x) \tag{1b}$$

$$C = L_{peak}/L_{rms} \tag{1c}$$

with x the amplitude vector representing the mono audio file associated with the track. For a stereo track $x = [x_L\ x_R]$, these equations become:

$$\begin{aligned} L_{rms} &= \frac{\sqrt{\frac{1}{N}\sum_{n=1}^{N}|x_L(n)|^2} + \sqrt{\frac{1}{N}\sum_{n=1}^{N}|x_R(n)|^2}}{2} \\ &= \frac{L_{rms,L} + L_{rms,R}}{2} \end{aligned} \tag{2a}$$

$$\begin{aligned} L_{peak} &= \max(\max(x_L), \max(x_R)) \\ &= \max(L_{peak,L}, L_{peak,R}) \end{aligned} \tag{2b}$$

$$C = L_{peak}/L_{rms} \tag{2c}$$

Additionally, a hysteresis gate determines which parts of the track are active (Figure 2):

$$a(n) = \begin{cases} 0, & \text{if } a(n-1) = 1 \text{ and } \tilde{x}(n) \leq T_1 \\ 1, & \text{if } a(n-1) = 0 \text{ and } \tilde{x}(n) > T_2 \\ a(n-1), & \text{otherwise} \end{cases} \quad (3)$$

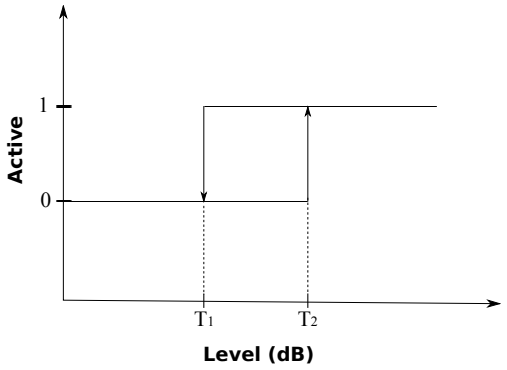

Fig. 2: Activity in function of audio level (hysteresis gate) following equation (3).

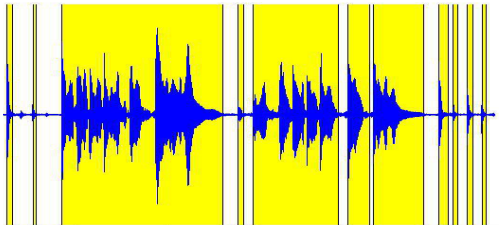

Fig. 3: Active audio regions highlighted as defined by the hysteresis gate.

where a is the binary vector indicating whether the track is active, $\tilde{x}$ a smoothed version of the track's audio, T_1 is the level threshold when the gate is off (audio is active), T_2 is the threshold when the gate is on (audio is inactive), and $T_1 \leq T_2$. For stereo tracks, x is summed to mono and divided by two.

Based on this definition, the following extra quantities are also included as metadata: the percentage of time the track is active, and the RMS level, peak level, crest factor and loudness when active.

Note that at this point no spectral information is extracted.

2.3. Processing modules

Research about the suggested order of processing is ongoing, and most practical literature bases the preferred order on workflow considerations [14, 15]. In some cases, at least one EQ stage is desired before the compressor, because an undesirably heavy low end or a salient frequency triggers the compressor in a way different from the desired effect [14]. For our purposes, we assume and ensure that the signal has no such spectral anomalies that significantly affect the working of the compressor (as confirmed by a short test). Instead, we place a high-pass filter before the compressor (preventing the compressor from being triggered by unwanted low frequency noise), and an equaliser after the compressor.

It is widely accepted that the faders and pan pots should manipulate the signal after the insert processors such as compressor and equaliser, and we place the pan pots after the faders as this is how mixing consoles are generally wired. Furthermore, because of the linear nature of these processes and their independence in this system, the order is of no importance in this context. Note however that the system allows for any order of processors.

Based on these considerations, the following order of processors is used for the assessment of this system: high-pass filter, dynamic range compressor, equaliser, fader and panner.

At this point, time-based effects are not incorporated in the system.

2.3.1. Dynamic range compression

We include a very generic compressor model, with a variable threshold layout (as opposed to for example a fixed threshold, variable input gain design), a quadratic knee and the following, standard parameters: threshold, ratio, attack and release ('ballistics'), and knee width [23].

Make-up gain is not used in this work since the levels are set at a later stage by the 'fader' module, which makes manipulating the gain at the compressor stage redundant. For now, there is also no side-chain filter, a side-chain input for other channels than the processed one, or lookahead functionality. The compressor processes the incoming audio sample per sample.

Stereo files (such as an overhead microphone pair) are compressed in 'stereo link' mode, i.e. the levels of both channels are reduced by an equal amount.

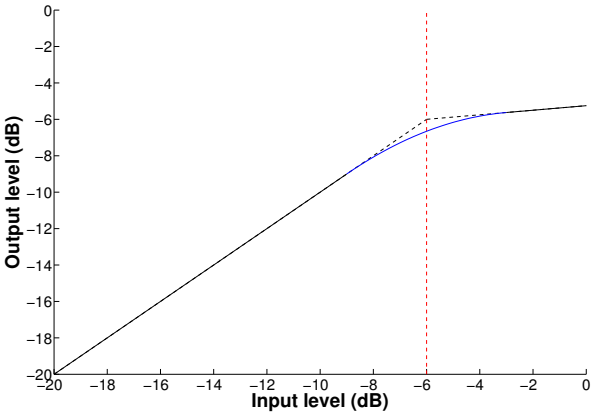

Fig. 4: Dynamic range compressor transfer function (with quadratic knee). Settings used here are: a 8:1 ratio, a -6 dB threshold and a knee width of 6 dB.

Practical literature lists a fair amount of suggested compressor settings for various instruments and various desired effects.

2.3.2. Equalising and filtering

A second essential processing step is the equalisation and filtering of the different tracks, or groups of tracks. Two tools take care of this task in this system: a high pass filter (implementing rules such as high pass filter with cutoff frequency of 100 Hz on every track but the bass guitar and kick drum) and a parametric equaliser (with high shelving, low shelving and peak modes). The parameters for the latter are *frequency*, *gain*, and Q (quality factor).

We use a simple biquadratic implementation for both the high-pass filter (12 dB/octave, as suggested by [21]) and the equaliser (second order filter per stage, i.e. one for every *frequency*/Q/*gain* triplet) [24].

Most rules found in practical literature are stated so that a great deal of interpretation can be given to them. Usually, an approximate frequency around which the track should be boosted or cut, but exact gain and quality factor values are absent. In this case, we try to estimate the gain (±3 dB is a generic gain value that seemed to work well during pilot tests, unless it is explicitly specified that the cut/boost should be modest or excessive) and the quality factor (sources often suggest to cut/boost a frequency region, such as 1-2 kHz, in which case the quality factor is chosen so that the width of the peak corresponds loosely with the width of this region).

When attempting to translate vague equalising suggestions into quantifiable mix actions, it helps to translate terms like 'airy', 'muddy' and 'bottom' into frequency ranges. This is possible because many sources provide tables or graphs that define these words in terms of frequencies [14–17].

2.3.3. Panning

The panning value is stored in the metadata of every track and initially set to zero. The value ranges from −1 (panned completely to the left) to +1 (panned completely to the right), and determines the the relative gain of the track during mixdown in the left versus the right channel.

Although we provide the option to choose from a variety of *panning laws*, for our purposes we use the -3 dB, equal power, sine/cosine panning law (different names can be found in literature), as it is the one that is most commonly used according to the practical audio engineering literature [14].

The gain of the left (g_{Li}) and right channel (g_{Ri}) for track i is then calculated as follows, with panning value p:

$$g_{Li} = \cos\left(\frac{\pi\,(p+1)}{4}\right) \tag{4}$$

$$g_{Ri} = \sin\left(\frac{\pi\,(p+1)}{4}\right) \tag{5}$$

Note that constant power is in fact obtained, regardless of the value of p, as $g_{Li}^2 + g_{Ri}^2 = 1$ (see Figure 5).

There is a lot of information available in practical literature on 'standard' panning values for every common instrument, both exact panning values as well as rules of thumb (e.g. describing the spread of harmony instruments over the stereo panorama).

2.3.4. Level

Like with panning, the 'level' variable per instrument is stored as metadata with the track. Its initial value being 0 dB, it can then be manipulated

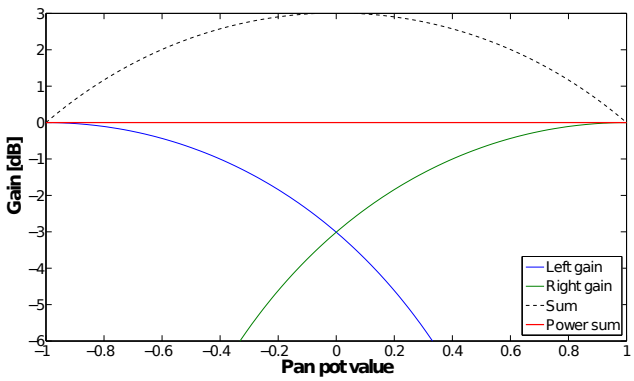

Fig. 5: Panning law: 3 dB equal power sine-law.

following the rule base (in absolute or relative terms - i.e. 'set level at x dB' or 'increase/decrease level by x dB') and applied during mixdown.

Except for vague guidelines ("every instrument should be audible", "lead instruments should be roughly x dB louder"), there is very little information available on exact level or loudness values from practical mixing engineering literature.

In this system, we start from a mix where all tracks have equal loudness, and then - for example - bring up those instruments where literature suggests a level boost, and bring down the instruments that should play a less prominent role such as ambience microphones.

2.3.5. Mixdown

The drum bus mixdown equation ((6) & (7)) and the total mixdown equation ((8) & (9)) then become:

$$d_L = \sum_{i=1}^{N_{drum}} 10^{\frac{L_i}{20}} \cdot g_{Li} \cdot x'_i \tag{6}$$

$$d_R = \sum_{i=1}^{N_{drum}} 10^{\frac{L_i}{20}} \cdot g_{Ri} \cdot x'_i \tag{7}$$

$$y_L = \sum_{j=1}^{N'} 10^{\frac{L_j}{20}} \cdot g_{Lj} \cdot x'_j + d'_L \tag{8}$$

$$y_R = \sum_{j=1}^{N'} 10^{\frac{L_j}{20}} \cdot g_{Rj} \cdot x'_j + d'_R \tag{9}$$

with $y = [y_L\ y_R]$ the stereo output signal, N_{drum} the number of drum tracks, N' the number of remaining tracks, $d = [d_L\ d_R]$ the drum submix (or drums *stem*), d' the processed drum submix after possible drum bus compression and equalisation, x'_i the processed audio of track i after possible compression and equalisation, and g_{Li} and g_{Ri} the left and right channel gain for track i (see above). Note that after the mixdown stage, y can still be processed by mix bus compression and equaliser.

3. LISTENING TEST

3.1. Test audio

To assess the performance of the system, we compare its output to mixes by two human mixing engineers, a plain, monophonic sum of the (normalised) input audio, and a completely automatic mix by processors based on existing automatic mixing algorithms.

For this experiment, the rule base is based on practical audio engineering [14–21] and Logic Pro 9 Channel EQ and Platinum Compressor presets.

Mixing engineer 1 (*'pro 1'*) has a professional experience spanning 12 years. Mixing engineer 2 (*'pro 2'*) has 3 years of professional mixing experience. For maximum comparability with the knowledge-engineered automatic mixing system (*'KEAMS'*), they are instructed to limit themselves to using a simple compressor, equaliser, pan pots and faders, and not to use automation (static settings). They can also process the drum bus and mix bus with a simple compressor and equaliser. Every song is mixed within 45 minutes or less. Note that no time-based effects like reverb are used, to allow for better comparison with the automatic mixing systems that lack this.

In the case of the existing automatic mixing algorithms, we use an automatic single-track compressor per track, a multitrack automatic equaliser, panner [3, 6] and fader [2, 4] (on the drum mix as well as on the total mix), and the single-track compressor and a single-track master EQ [11] on the mix bus. The processors are implemented in the form

of VST (Virtual Studio Technology) effect plugins in Reaper, a DAW capable of accommodating multichannel audio and plugins. Because the mix settings are adjusted during playback (real-time cross-adaptive audio effects), the audio is played back once before rendering the mix to allow the parameters to converge to suitable initial values. Note that in this case, the VST system (*'VST'*) is unaware of the functions of the different tracks. It does not know which tracks are part of the drum set, or which are lead and which are background instruments. Instead, it extracts dynamic and spectral information in real-time and adapts the mixing parameters based on these values.

We used publicly available raw audio tracks from Shaking Through, an online music project by Weathervane Music [25]. The five songs used in this experiment range from light pop-rock to heavier alternative rock. For every song, only one channel is selected per instrument (two in the case of instruments recorded in stereo), as sometimes the raw tracks include multiple recordings of the same instrument by different microphones and/or via direct injection.

To minimise processing time, avoid drastic dynamic and spectral variations (since the mixing applied parameters are static in the current implementation, see above), and make the listening tests as well as the manual mixes not too demanding, all input tracks are just 4 bars long. This yields audio files between 11 and 24 seconds. The number of tracks varies from 10 to 22. Every song contains at least vocals, bass, guitar and or keyboards, kick drum, snare drum, and drum overhead microphones.

The levels of the resulting mixes are then adjusted to obtain equal loudness, following the ITU loudness standard [26], to remove bias towards louder (or softer) samples during the listening test.

3.2. Test design

The listening test method we used was a multiple stimulus test with hidden anchor, expected to yield accurate results while minimising the subjects' time and effort [27]. This corresponds with a MUSHRA test [28] (multiple stimulus with hidden anchor and reference) except that in this case there was no reference. The hidden anchor here is a monophonic sum of the raw audio, where every channel has been normalised. However, it is uncertain whether even high-performing subjects will consistently rate this anchor the lowest, as it is possible that other mixes are perceived to be poor as well, or that the monophonic sum without processing at mixdown is an acceptable mix for some songs. Figure 6 shows the interface used in this experiment. From here on, the position on the scale is represented as a value from 0 ('Bad') to 100 ('Excellent').

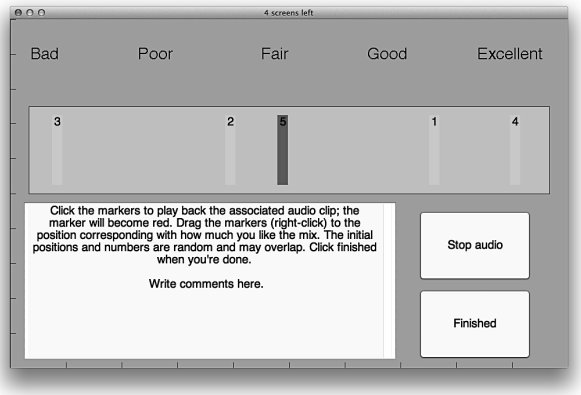

Fig. 6: Interface used during the listening experiment. Every marker corresponds to a different version of the song fragment. The highlighted marker represents the fragment that is currently playing. To avoid excessive focus on the very first few seconds of the fragment, subjects are able to toggle between mixes as they play, while also having the possibility of stopping the audio entirely and play back any sample from the beginning.

The order of the songs, and the order and numbering of the versions per song is randomised, to avoid any kind of bias or subject performance difference related to the order of playback.

The listening tests were conducted in a dedicated, well-isolated listening room, using an Apogee Duet audio interface and Beyerdynamic DT770 Pro headphones (closed, circum-aural). Figure 7 shows the transfer function of this pair of headphones, inevitably influencing the perceived sound during the test, but more controlled than other listening environments at our disposal.

A total number of 15 subjects participated in the listening experiment. Two thirds of the subjects were

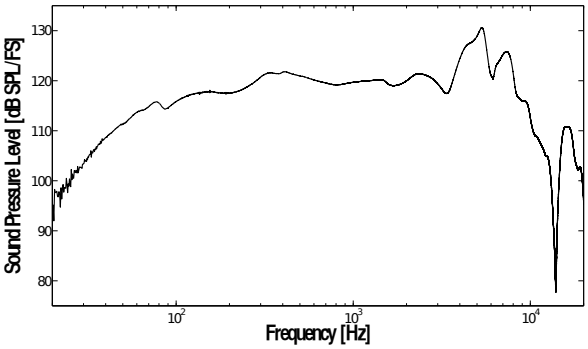

Fig. 7: Transfer function of the set of headphones used for the listening test, as measured using a KEMAR artificial head and sine sweep excitation. It is an average of three left and right channel recordings, and shows the SPL at 0 dBFS as a function of frequency.

male. 7 of the 15 subjects had at least some practical audio engineering experience (mixing and/or recording). All had previously participated in listening tests, and played musical instruments for at least 5 years (although neither of these were prerequisites to participate in the test).

The subjects were asked to rate the five different versions of the same song fragment according to 'sound' (rather than 'mix', which may have encouraged subjects to focus on specific mix aspects rather than rate their affective impression, or 'quality', which may have caused subjects to look for data compression artefacts). The subjects did not have any information about the audio content or the research goal before taking the test. The complete task took the subjects 15 minutes 52 seconds on average, with a standard deviation of 4 minutes 51 seconds (with total times ranging from 7 minutes 52 seconds to 26 minutes 34 seconds). The time per song did not depend much on which song was being assessed, but did decrease significantly from one trial to the next (from 4 minutes 31 seconds for the first song to 2 minutes 31 seconds for the last song). It should be noted that the first trial typically included a brief demonstration of the user interface.

After the test, their overall impression and points of focus were determined during an informal chat with each subject.

4. RESULTS AND DISCUSSION

Figure 8 shows the ratings for each mixing system, for each song. A few trends are immediately apparent: the monophonic sum is (not surprisingly) generally rated worse than the other mixes, and the fourth song (heavy rock) is consistently rated worse than the other songs.

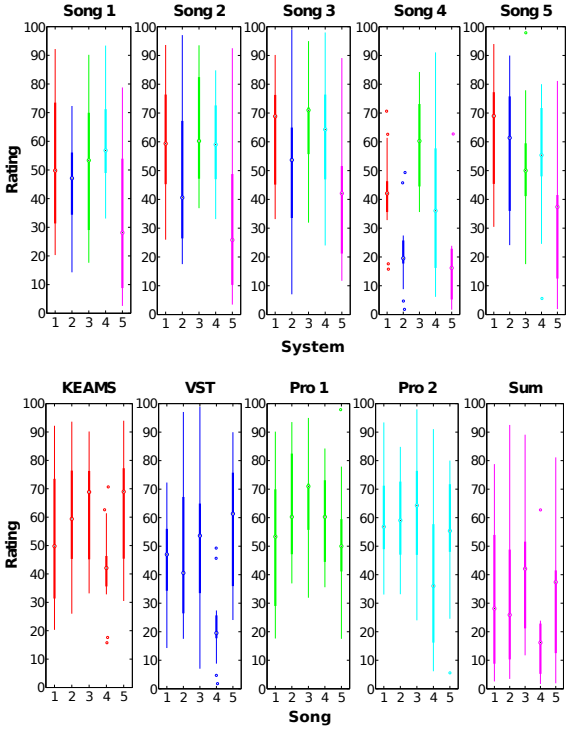

Fig. 8: Box plot representation of the ratings per song and per system (1: 'KEAMS', 2: 'VST', 3: 'pro 1', 4: 'pro2', 5: 'sum'). Following the classic definition of a box plot, the dot represents the mean, the bottom and top of the 'box' represent the 25% and 75% percentile, the vertical lines extend from the minimum to the maximum when smaller than 1.5 times the 25% and 75% percentiles, and the outliers are represented by open circles.

To quantify the effect of the mixing systems (and of the songs), we conducted an analysis of variance (ANOVA). The system and song effect sizes are

$R^2_{system} = 0.17$ (large effect [29]) and $R^2_{song} = 0.09$ (medium effect).

Instead of rejecting the null hypothesis (in which we are unsuccessful), we now want to investigate the pairwise differences between the mixing systems. Rather than proving that all mixing systems perform significantly different, we could at least find that system A performs significantly better than system B. To this end, we perform a multiple comparison of the population marginal means (a Bonferroni test with tolerance of 0.05, see Figure 9 and the effect sizes in Table 1), looking at the pairwise rather than the familywise error rate.

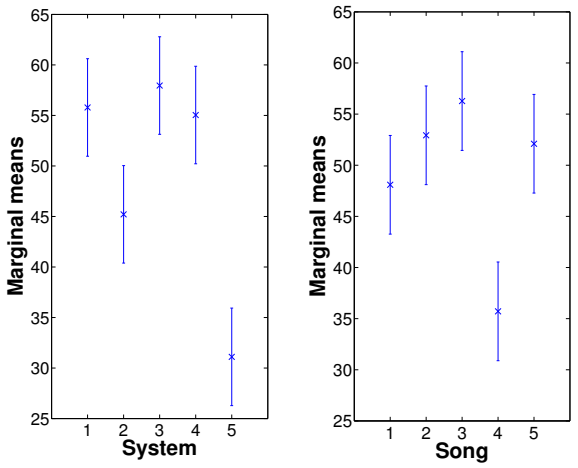

Fig. 9: Multiple comparison of population marginal means showing the effect of system (1: 'KEAMS', 2: 'VST', 3: 'pro 1', 4: 'pro2', 5: 'sum') and song.

We learn that the normalised sum of the raw audio indeed performs notably worse than the other mixes (with a large effect size). The same is true for the fourth song compared to all other songs (with a medium effect size for all but the third song, which is rated the highest and differs from the fourth song with a large effect size). Furthermore, the automatic mix is rated lower than the human mixes and the rule-based system (with a medium effect size for engineer '*pro 1*' and a small effect size for the others). No significant difference between the rule-based system and the human mixing engineers is revealed by this experiment.

Systems	R^2	Songs	R^2
1 2	.053	1 2	.010
1 3	(.003)	1 3	.032
1 4	(.000)	1 4	.068
1 5	.238	1 5	.007
2 3	.079	2 3	.005
2 4	.044	2 4	.117
2 5	.080	2 5	.000
3 4	(.005)	3 4	.172
3 5	.281	3 5	.008
4 5	.218	4 5	.113

Table 1: Effect sizes of pairwise differences of ratings (1: 'KEAMS', 2: 'VST', 3: 'pro 1', 4: 'pro2', 5: 'sum'). Where the difference is not significant ($p > \frac{0.05}{10} = 0.005$, with the number of pairwise comparisons $= 10 = \frac{5!}{(5-2)!(2!)}$), the effect size is shown between brackets.

During the subsequent conversation, all subjects claimed to partly or entirely judge the different mixes based on the balance in level of the sources, and/or the audibility and masking of instruments. Examples of balance issues include overpowering (backing) vocals, a barely audible lead vocal, and sometimes inaudible instruments like a guitar or a piano. In general, the '*sum*' (peak-normalising all sources without any other processing may cause a bad balance) and '*VST*' (making no distinction between lead and background instruments) appeared to cause these remarks. However, it should be noted mixing engineer '*pro 1*' sometimes chose to omit (mute) an instrument as an artistic choice (an option mixing engineers often gladly use [19]), more specifically a guitar in Song 4 and a piano in Song 5. This didn't always go unnoticed, although it seemed this was often perceived as a good thing.

Many (9 out of 15) reported 'spacing', 'location' or 'panning' to be of influence in their ratings, sometimes referring to 'weird panning' (found to relate to the '*VST*' system that sometimes panned the snare drum or lead vocals considerably to the left or right side, which is unconventional and rarely desired) and sometimes to the '*sum*' where all instruments are 'centred', which was often a bad thing although some found this to work well with certain songs.

Other remarks included: an overly harsh guitar

sound with one version of Song 4 (presumably the '*KEAMS*' version, where default guitar EQ settings are applied to already quite bright guitars), a lack of blend (associated with the lack of reverb) and the absence of context (preferences may have been different had the fragment been part of a bigger whole). Overall, there seemed to be a tendency to focus on the vocals: 10 out of 15 explicitly mentioned vocals in either a balance or spatial context.

5. CONCLUSIONS AND FUTURE PERSPECTIVE

The results of this experiment and the subsequent conversation with the subjects suggest a good performance of the knowledge-engineered system, with no significant difference from human mixes. Moreover, it outperforms the automatic system that does not take semantic information into account, even though it uses less sophisticated feature extraction.

At the same time, an important shortcoming was highlighted during the post-experiment discussion with the subjects: the system assumes particular spectral and dynamic characteristics, which causes problems when the recorded signals deviate from this. Similarly, it should be noted that whereas the raw audio tracks used for this test were of high quality, it is doubtful whether the system will perform well when the input audio is of low quality or at least less than conventional when it comes to dynamic and spectral characteristics.

For this reason, we believe this system can be vastly improved by expanding the set of measurement modules, to allow for more enhanced listening and processing, such as detecting and resolving inter-channel masking. This means effectively moving towards a more hybrid system, where semantic rules (processing dependent on high-level semantic information such as instrument tags) and more advanced, cross-adaptive signal processing (processing dependent on signal features of the track itself as well as other tracks) are combined to obtain the highest possible performance.

A second important research direction is the perceptual motivation (or disproof) of the rules found in practical audio engineering literature. The developed system proves to be a suitable framework for investigating user preferences of different mixing approaches and settings, as it allows for easy comparison of different sets of rules, different processor implementations and the order of processors.

Formalisation of the rule list into a tractable knowledge base will allow efficient handling in description logic contexts, facilitate the expansion and editing of the rule base and enable sharing of rule sets.

Finally, in order to obtain acceptable mixes automatically, it will be necessary to incorporate time-based effects such as reverberation and delay in the system. Further research is necessary to include a viable autonomous reverb/delay processor and establish reverberation rules.

The test audio used for this experiment is available on www.brechtdeman.com.

6. ACKNOWLEDGEMENTS

The authors would like to thank mixing engineers Pedro Duarte Pestana and Gauthier Grandgirard, as well as everyone who participated in the listening test.

7. REFERENCES

[1] D. Dugan, "Automatic microphone mixing," *Journal of the Audio Engineering Society*, vol. 23, 1975.

[2] S. Mansbridge, S. Finn, and J. D. Reiss, "Implementation and evaluation of autonomous multitrack fader control," *132th convention of the Audio Engineering Society*, April 2012.

[3] S. Mansbridge, S. Finn, and J. D. Reiss, "An autonomous system for multi-track stereo pan positioning," *133rd Convention of the Audio Engineering Society*, October 2012.

[4] E. Perez-Gonzalez and J. D. Reiss, "Automatic gain and fader control for live mixing," *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, October 2009.

[5] E. Perez-Gonzalez and J. D. Reiss, "An automatic maximum gain normalization technique with applications to audio mixing," *124th Convention of the Audio Engineering Society*, May 2008.

[6] E. Perez-Gonzalez and J. D. Reiss, "Automatic mixing: Live downmixing stereo panner," *Proc. of the 10th Int. Conference on Digital Audio Effects*, September 2007.

[7] E. Perez-Gonzalez and J. D. Reiss, "Automatic equalization of multi-channel audio using cross-adaptive methods," *127th Convention of the Audio Engineering Society*, October 2009.

[8] E. Perez-Gonzalez and J. D. Reiss, "Improved control for selective minimization of masking using inter-channel dependency effects," *Proc. of the 11th Int. Conference on Digital Audio Effects*, September 2008.

[9] J. D. Reiss, "Intelligent systems for mixing multichannel audio," *17th International Conference on Digital Signal Processing (DSP),* pp. 1 –6, July 2011.

[10] M. J. Terrell and J. D. Reiss, "Automatic monitor mixing for live musical performance," *Journal of the Audio Engineering Society*, Volume 57, Issue 11, pp. 927-936; November 2009.

[11] Z. Ma, J. D. Reiss, and D. A. A. Black, "Implementation of an intelligent equalization tool using Yule-Walker for music mixing and mastering," *134th Convention of the Audio Engineering Society*, May 2013.

[12] J. Scott, M. Prockup, E. Schmidt, and Y. Kim, "Automatic multi-track mixing using linear dynamical systems," *Proceedings of the 8th Sound and Music Computing Conference, Padova, Italy*, 2011.

[13] J. Scott and Y. Kim, "Analysis of acoustic features for automated multi-track mixing," *Proceedings of International Society for Music Information Retrieval (ISMIR)*, 2011.

[14] R. Izhaki, *Mixing audio: Concepts, Practices and Tools*. Focal Press, 2008.

[15] B. Owsinski, *The Mixing Engineer's Handbook*. Course Technology, 2nd ed., 2006.

[16] K. Coryat, *Guerrilla Home Recording: How to Get Great Sound from Any Studio (no Matter how Weird Or Cheap Your Gear Is)*. MusicPro guides, Hal Leonard Corporation, 2008.

[17] D. Gibson, *The Art Of Mixing: A Visual Guide To Recording, Engineering, And Production*. Thomson Course Technology, 2005.

[18] P. White, *Basic Effects & Processors*. The Basic Series, Music Sales, 2000.

[19] A. Case, *Mix Smart: Professional Techniques for the Home Studio*. Focal Press, Taylor & Francis, 2011.

[20] A. Case, *Sound FX: Unlocking the Creative Potential of Recording Studio Effects*. Taylor & Francis, 2012.

[21] M. Senior, *Mixing Secrets*. Taylor & Francis, 2012.

[22] T. Wilmering, G. Fazekas, and M. Sandler, "Towards ontological representations of digital audio effects," *Proceedings of the 14th Int. Conference on Digital Audio Effects (DAFx-11)*, September 2011.

[23] D. Giannoulis, M. Massberg, and J. D. Reiss, "Digital dynamic range compressor design - a tutorial and analysis," *Journal of the Audio Engineering Society*, vol. 60, no. 6, pp. 399–408, 2012.

[24] http://www.musicdsp.org/files/Audio-EQ-Cookbook.txt

[25] http://www.shakingthrough.com

[26] International Telecommunication Union, *"Algorithms to measure audio programme loudness and true-peak audio level,"* ITU-R BS.1770-2, 2011.

[27] B. De Man and J. D. Reiss, "A pairwise and multiple stimuli approach to perceptual evaluation of microphone types," in *134th Convention of the Audio Engineering Society*, May 2013.

[28] International Telecommunication Union, *"Multiple Stimuli with Hidden Reference and Anchor,"* ITU-R BS. 1534-1, 2003.

[29] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, 2nd edition. Routledge, 1988.