DEVELOPMENT OF A QUALITY ASSURANCE AUTOMATIC LISTENING MACHINE (QUAALM)

Daniel J. Gillespie *
Newfangled Audio
New York, NY
DGillespie@newfangledaudio.com

Woody Herman
Eventide Inc.
Little Ferry, NJ
WHerman@eventide.com

Russell Wedelich
Eventide Inc.
Little Ferry, NJ
RWedelich@eventide.com

ABSTRACT

This paper describes the development and application of a machine listening system for the automated testing of implementation equivalence in music signal processing effects which contain a high level of randomized time variation. We describe a mathematical model of generalized randomization in audio effects and explore different representations of the effect's data. We then propose a set of classifiers to reliably determine if two implementations of the same randomized audio effect are functionally equivalent. After testing these classifiers against each other and against a set of human listeners we find the best implementation and determine that it agrees with the judgment of human listeners with an F1-Score of 0.8696.

1. INTRODUCTION

When Eventide began development of their new H9000 effects processor a primary task was the porting of a large signal processing code base from the Motorola 56000 assembly language to C++ targeted to several different embedded processors. While industry standard cancellation testing was possible for deterministic effects, most of the effects involved some amount of random time variation causing test results to vary significantly from the judgment of human listeners. This created a problem when ascertaining that the thousands of presets used in the H8000 were working correctly on the H9000. To solve this problem we developed a machine listening system to replace the first stage of human listening to determine if two different implementations sound equivalent.

Comparing the similarity of two audio signals is done in many contexts. The field of audio watermarking aims to embed a unique code in an audio stream such that the code is imperceptible to the human ear and is recoverable after the application of several different auditorily transparent transformations such as time scale modification [1][2]. Several different techniques are used to quantify how perceptible the added code is to the listener. These include the Perceptual Audio Quality Measure (PAQM), the Noise to Mask Ratio (NRM), and the Perceptual Evaluation of Audio Quality (PEAQ) [3]. However, while these methods aim to make a perceptual comparison between two recordings, the comparison is primarily interested in determining if two signals sound exactly the same. Unfortunately, these comparisons are not useful to us as we need to determine if two sets of unique signals are generated by the same random process, even when these sets of signals vary significantly inside each set.

There have also been several attempts to reverse engineer the specific settings of audio effects [4][5][6], however these systems are often aimed at specific effects types and rely on a specific model. At the most general, Barchiesi and Reiss assume that the effects are linear and time-invariant[4], which we cannot assume here.

The fields of audio query by example and music recommendation systems have a looser definition of audio similarity which might have some applications to our problem. A comparison of some systems which aim to define this more general similarity is given by Pampalk et al in [7]. A particularly interesting implementation from Helén et al [8] uses a perceptual encoder followed by a compression algorithm to determine similarity by the compression ratio of the signals separately vs combined. Another interesting probabilistic approach is given by Virtanen [9]. These models may have the capacity to model the variation we expect between and within the sets of signals we are evaluating, however, their implementations are very resource intensive and the nature of these problems is that there is no ground truth by which to compare the algorithms to each other. We suspect that we might be able to solve our problem with a more compact model.

In Section 2 we give a deeper description of the problem. Section 3 describes the theory behind our solution. Section 4 mentions some practical considerations of our system. Section 5 describes the validation experiments we ran, and Section 6 gives results of these experiments and some analysis. Section 7 gives a brief conclusion, Section 8 some acknowledgments, and Section 9 includes references.

2. PROBLEM DESCRIPTION

When testing whether two audio signal processing implementations are functionally equivalent it is common to send a test signal $x(t)$ through each of them and compare the output signals $y_1(t)$ and $y_2(t)$ [10]. If the magnitude difference $z(t)$ between these two outputs is less than 2η, for some very small η, the two implementations can be considered equivalent.

This can be written as

$$y_n(t) = f_n(x(t)) \tag{1}$$

$$z(t) = |y_1(t) - y_2(t)|, \tag{2}$$

$$z(t) < 2\eta; \forall t \in T, \tag{3}$$

where $f_n(x)$ represents the n^{th} distinct implementation being tested.

This test works well for musical effects that can be characterized by a deterministic system plus a small residual, as long as the system memory is less than the measurement time scale T. This is true because this test relies on the the implicit signal model

$$f_n(x(t)) = f(x(t)) + \eta N_n(t) \tag{4}$$

* For Eventide Inc.

where $f(x(t))$ is the theoretically perfect implementation and $N_n(t)$ is a continuous random process on the range $[-1, 1]$ representing the implementation-specific variation from $f(x(t))$. The test in Equation (3) places no constraints on $N_n(t)$ other than that it is strictly bounded to the range $[-1, 1]$, nor do we assume any. It is likely that some realizations of $N_n(t)$ produce more audible effects than others, but in practice the cancellation tests works by keeping η very small.

A very simple example of a deterministic effect that uses this signal model is a gain effect with a constant gain of g. This system can be represented as

$$f(x(t)) = gx(t) \tag{5}$$

When implemented digitally it becomes

$$f_n(x(t)) = g[x(t) + q_n(t)] + t_n(t) \tag{6}$$

Where $q_n(t)$ represents the quantization noise and $t_n(t)$ represents the truncation error of the multiply, each in the n^{th} implementation. Both of these types of error can be modeled by additive noise, and with a slight re-arrangment we can show that it fits the signal model of Equation (4)

$$f_n(x(t)) = gx(t) + gq_n(t) + t_n(t) \tag{7}$$

$$f_n(x(t)) = f(x(t)) + \eta N_n(t) \tag{8}$$

$$|f_1(x(t)) - f_2(x(t))| = |\eta N_1(t) - \eta N_2(t)| \tag{9}$$

and as long as $|gq_n(t) + t_n(t)| < \eta$ is strictly true for two different implementations, we can call them equivalent using the cancellation test in Equation (3).

A slightly more complicated deterministic process which still passes the test in Equation (3) is the simple tremolo effect. To create a tremolo effect we define g as a function of time, now the system can be described by

$$g(t) = sin(w_0 t) \tag{10}$$

$$f(x(t)) = sin(w_0 t)x(t) \tag{11}$$

By adding an additional $s_n(t)$ term to represent the error in generating the sine wave, we can write an implementation-specific version.

$$f_n(x(t)) = (sin(w_0 t) + s_n(t))(x(t) + q_n(t)) + t_n(t) \tag{12}$$

After grouping the additive components we can still factor the implementation specific components into a simple additive term.

$$f_n(x(t)) = sin(w_0 t)x(t) + \eta N_n(t), \tag{13}$$

$$\eta N_n(t) = sin(w_0 t)q_n(t) + s_n(t)(x(t) + q_n(t)) + t_n(t). \tag{14}$$

As long as the implementation errors $q_n(t)$, $s_n(t)$, and $t_n(t)$ are kept small enough the two implementations can be considered equivalent, as is demonstrated by

$$|f_1(x(t)) - f_2(x(t))| = \eta |N_1(t) - N_2(t)|. \tag{15}$$

When considering the addition of intentional randomization to these types of effects it might be tempting to consider them to be additive processes using the existing signal model. For instance, if we consider the implementation error of the sine wave $s_n(t)$ in equation (12) to instead represent an intentional random process integral to the sound of the effect, we might consider trying to reduce it's contribution to the measurement error by averaging over several recordings of the effect. If all sources of randomization can be considered purely additive, we could take several measurements of each implementation and compare the mean and variance of each to make a determination which may prove sufficient to call them functionally equivalent.

Unfortunately, many effects implement processing whose randomization cannot be treated as simple additive noise. For these effects even a relaxation of the cancellation test in Equation (3) is not appropriate because the signal model used does not have the capacity to discriminate between equivalent implementations and ones that differ. We can show an example of this by adding a random initial phase θ_n to our tremolo example and deriving the equation for the cancellation test. If we treat θ_n as a uniformly distributed random variable between $-\pi$ and π the system can be described by

$$f_n(x(t)) = (sin(w_0 t + \theta_n) + s_n(t))(x(t) + q_n(t)) + t_n(t) \tag{16}$$

After grouping the additive components we are not able to factor the theoretically perfect $f(x(t))$ out of the model.

$$f_n(x(t)) = sin(w_0 t + \theta_n)x(t) + \eta N_n(t), \tag{17}$$

and therefore the difference will depend on θ_n

$$|f_1(x(t)) - f_2(x(t))| = |(sin(wt + \theta_1)x(t) - sin(wt + \theta_2)x(t)) + (\eta N_1(t) - \eta N_2(t))| \tag{18}$$

Even if we make the additive sources of error very small, if we cannot control the distribution of θ_n there will still be a large source of error determined by this initial condition. This might seem to be a contrived comparison, but consider that the two implementations might be running on different hardware, or even implemented in an analog circuit. In these situations we may not be able to synchronize θ_n. More complicated effects often have several processes with randomized initial conditions which we will store in the vector Θ_n and may even be the result of one or more random processes $\Psi_n(t)$. We can write this as the more general formulation

$$y_n(t) = f_n(x(t), \Psi_n(t), \Theta_n) \tag{19}$$

In these instances, $\Psi_n(t)$ and Θ_n may not always be able to be controlled between various implementations, and when they are not the same we can expect the test in Equation (3) to fail.

However, in practice we found that both in-house testers and end users are able to reliably detect similarity or difference in these categories of randomized time-variant effects, even when sources of randomization are not controlled. Therefore we hoped to design a machine listening algorithm to make these implementation equivalence determinations in much the same way that humans do. The goal was to create an algorithm that would reliably detect functional equivalence between implementations of both the easily tested deterministic, as well as the highly randomized audio effects without any prior knowledge about the effect under test.

3. THEORY

To extend the simple model in Equation (3) to systems with undefined random variations we will form a probabilistic interpretation

and use several samples of each implementation $f_n(\cdot)$ to learn a model of its behavior. Once we learn models for each $f_n(\cdot)$ we can compare these models to make a decision about the similarity of the implementations.

3.1. Representing the Data

As is done in the classical problem description above, we chose to represent each $f_n(\cdot)$ by testing the appropriate response $y_n(t)$ to the common test signal $x(t)$. Therefore the learning problem happens on vectors representing the $y_n(t)$ signals rather than the $f_n(\cdot)$ processes directly. Because of this, the result of the test depends both on the test method as well as the forcing signal $x(t)$. Specifically, in the case of linear time-invariant systems, $y_n(t)$ must be long enough to capture the entire expected impulse response of $f_n(\cdot)$, and it must have sufficient bandwidth to measure the expected bandwidth of $f_n(\cdot)$. When the system is expected to be nonlinear it must have sufficient changes in amplitude to capture these effects. Finally, when the system is expected to be time-variant, $y_n(t)$ must be long enough to capture this time-variance, or there must be enough samples of $y_n(t)$ to sufficiently span the expected range of variance.

So given the test signal $x(t)$, we will now learn a model Y_n for each implementation $f_n(\cdot)$ representing its response $y_n(t)$.

To learn the Y_n we must represent the signal $y_n(t)$ as a fixed length feature vector $\mathbf{y}_n$, where each discrete sample in $\mathbf{y}_n$ will be a feature in our probabilistic model. For these representations we consider several choices. We can choose to simply use the discretized samples $y_n[i]$ directly. This has the benefit of being similar to the tried and true method in Equation (3) and does not lose any data. Additionally, when using this representation, any purely additive or multiplicative differences between implementations of $f_n(\cdot)$ will result in purely additive or multiplicative differences in $\mathbf{y}_n$. Therefore, even if we assume the features of Y_n are strictly independent, Y_n will still be able to capture these differences. However, if $f_n(\cdot)$ has memory, errors in the part of the system with memory will be spread across several different features in $\mathbf{y}_n$ and we will have to include some dependency structure in Y_n to accurately model these differences, which can make the model more complicated.

We can also choose to take the magnitude FFT of $y_n[i]$, which has the added benefit of keeping most of the data when the FFT size is large. However, if we do so, convolutions in the $f_n(\cdot)$ process will become simple multiplications. Therefore a model that assumes independence will be more robust to small convolutive variations, like a small change in a delay time parameter, at the expense of being less robust to small multiplicative variations, like a small change in a gain coefficient.

A compromise solution might be to take the magnitude STFT of y_n and vectorize it. This would combine the benefits of both the sample-wise and FFT representations, while maintaining the locality of differences in both time and frequency. This might put a limit on the amount of dependency the model needs to assume, or the amount of error we'd expect to see by assuming independence.

Finally, considering we're looking to replace human listeners we might take a cue from speech recognition research [11], and choose to represent the audio as a vectorized set of MFCC coefficients. While this representation does throw away data, depending on the length of the test signal $x(t)$, losing this data may help us with regard to the "curse of dimensionality" [12], which describes how the modeling capacity of a given model can suffer as the dimensionality of the training data increases. In Experiment 1 we will try all four of these data representations to determine which gives the best classification accuracy across a number of effects.

3.2. Forming a Probabilistic Representation

When dealing with sources of variation it is often useful to build the model in terms of a probabilistic formulation. If we believe that M successive applications of $f_n(\cdot)$ to $x(t)$ will result in M distinct results $y_{mn}(t)$ and $\mathbf{y}_{mn}$ then we might choose to model the $\mathbf{y}_{mn}$ vectors as a random process from which we can draw samples. A commonly used generative model for high dimensional continuous data is the multivariate normal distribution with the assumption of independence between the dimensions [13]. This model has the benefit of fitting many naturally occurring processes, while also having a tractable calculation of the log likelihood that a particular sample has come from the underlying model. This is true even when used with very high dimensional feature vectors, like we expect to have here.

To do this, we define

$$Y_n \sim \mathcal{N}(\mu_n, \sigma_n^2)\,. \tag{20}$$

$$\mu_n = \frac{1}{M}\sum_{m=1}^{M} \mathbf{y}_{n\,m} \tag{21}$$

$$\sigma_n = \frac{1}{M-1}\sum_{m=1}^{M} (\mathbf{y}_{n\,m} - \mu_n)(\mathbf{y}_{n\,m} - \mu_n)^{\mathrm{T}}. \tag{22}$$

where $\mathcal{N}$ is the multivariate normal distribution, μ_n is the mean of the M samples of $\mathbf{y}_n$, and σ_n is the variance vector. In this instance the covariance matrix reduces to a vector because independent features imply that all non-diagonal elements of the covariance matrix are zero.

3.3. Making a Decision

Once we have a model Y_n representing the effect $f_n(\cdot)$ for each implementation n, we must compare them to measure their similarity. A common method of measuring the similarity between two probability distributions is the Kullback-Leibler divergence $D_{\mathrm{KL}}(P\|Q)$ [14]. The Kullback-Leibler divergence from Q to P measures the information gained when one modifies the prior distribution Q to the posterior distribution P.

When P and Q are distributions of a continuous random variable the Kullback-Leibler divergence is defined by the probability densities of P and Q, respectively $p(x)$ and $q(x)$, as

$$D_{\mathrm{KL}}(P\|Q) = \int_{-\infty}^{\infty} p(x)\,\log\frac{p(x)}{q(x)}\,\mathrm{d}x, \tag{23}$$

To determine the similarity of two implementation models Y_n we can choose the target implementation to represent the true distribution P and choose the implementation under test to represent the prior distribution Q. The smaller the information gained by modifying Q to match P, the more similar the distributions Q and P are considered. Therefore, by finding the Kullback-Leibler divergence $D_{\mathrm{KL}}(Y_{target}\|Y_{test})$ we can gain a measure of similarity between the known good implementation $f_{target}(\cdot)$ and the one being tested $f_{test}(\cdot)$.

When P and Q are both multivariate normal distributions $\mathcal{N}_P$ and $\mathcal{N}_Q$ with independent components the KL divergence can be simplified to:

$$D_{\mathrm{KL}}(\mathcal{N}_P \| \mathcal{N}_Q) = \frac{1}{2} \sum_D \ln |\sigma_Q| - \sum_D \ln |\sigma_P| - D + \sum_D \frac{\sigma_P}{\sigma_Q} + (\mu_Q - \mu_P)' \frac{1}{\sigma_Q} (\mu_Q - \mu_P) \quad (24)$$

This can be thresholded to make a hard decision, or be simply reported as an error measure.

Another option is to treat this decision as a classification problem by determining if it is more likely that a given sample $\mathbf{y}_n$ came from a model representing one specific implementation, or a joint model representing both representations. For a multivariate normal distribution with independent dimensions, the log likelihood can by calculated by

$$\ln(L) = -\frac{1}{2} \sum_{d=1}^{D} \frac{(x_d - \mu_d)^2}{\sigma_d^2} - \frac{D}{2} \ln 2\pi - \frac{1}{2} \sum_{d=1}^{D} \ln \sigma_d^2 \quad (25)$$

We then pull out one sample $\mathbf{y}_{n,m}$ and form two sets from the remaining samples. The set S_{test} represents the remaining samples from the implementation being tested, while the set S_{all} is formed from the union of S_{test} and S_{target}, which is the set of all samples from the implementations being tested against.

$$S_{test} = \mathbf{y}_{n,l}; \forall l \neq m \quad (26)$$

$$S_{all} = \mathbf{y}_{k,l}; \forall k \neq n \cup l \neq m \quad (27)$$

Now we build models Y_{test} from set S_{test} and Y_{all} from set S_{all} and calculate the log likelihoods, $\ln(L_{test})$ and $\ln(L_{all})$. If $\ln(L_{test}) > \ln(L_{all})$, then the sample being tested fits the samples that came from the test implementation better than it fits a collection of the samples from all implementations. This is an indication that the implementation being tested differs from the target. However, if $\ln(L_{all}) > \ln(L_{test})$, then the two implementations are similar enough that the held-out sample cannot be grouped specifically with the test implementation. This is an indication that the two implementations are functionally equivalent.

To reduce the effect of outliers in this decision making, we use a method similar to leave-one-out cross validation, and repeat this process for each m in M, then use a voting method [15] to determine if the implementations should be considered equivalent.

4. PRACTICAL CONSIDERATIONS

In practice the Quality Assurance Automatic Listening Machine must run relatively quickly on a large variety of presets with as little human intervention as possible. For the purposes of the following experiments, the test signal $x(t)$ was chosen to be 3 seconds of full bandwidth noise, followed by 3 seconds of silence, followed by a 3 second long logarithmic chirp, followed by a final 3 seconds of silence. This specific signal was chosen to reflect the particulars of the expected effects. While we believe that there are some effects for which this signal will be insufficiently long, a necessary trade off must be made to keep the dimensionality reasonable for the classifiers sake and to keep the recordings short for the sake of the test duration and for human validation. We didn't experiment with different test signals, though we believe that this might be a good route for further improvement. Additionally, we chose to build our models based on a sample of 10 recordings from each preset being tested.

In operation, we will also have a preference for false failures over false passes, because a false negative that allows a human listener to double check is preferable to a false positive that allows a potential implementation problem to go into production. For this reason, we will score the classification experiments using both the standard F Score as well as the Precision and Recall scores. Preference is given to the Precision score which penalize the algorithm only for false positives.

From [16] Precision is defined as

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \quad (28)$$

and Recall is defined as

$$\mathrm{Recall} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}, \quad (29)$$

where TP is the number of true positives, or instances where both the human listeners and the QuAALM decided that the implementations were the same, TN is the number of true negatives, or instances where both the human listeners and the QuAALM decided that the implementations were different, FP is the number of false positives, or instances where the QuAALM declared that the two implementations were the same, but the human listeners did not, and FN is the number of false negatives, or instances where the QuAALM declared that the two implementations were different, but the human listeners decided that they were the same.

Then, the F-measure is defined as the average of Precision and Recall:

$$\mathrm{F}_1 = 2 \cdot \frac{\mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \quad (30)$$

$$\mathrm{F}_1 = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \quad (31)$$

5. EXPERIMENTAL DESIGN

We ran two different experiments. The first is meant to find the best set of features and the best decision algorithm. The second is meant to validate the QuAALM machine by comparing it with human listeners.

5.1. Experiment 1: Determination of Best Signal Representation and Decision Algorithm

The first experiment intends to determine the best representation of the data between the options of the naive basis (samples), full magnitude FFT, magnitude STFT, and MFCC features. For each data representation the experiment finds the result using the the voting classifier method, as well as the thresholded Kullback-Leibler divergence. This test is run using 10 recordings from each of 19 individual effects which were hand picked to represent the diversity of expected effects. The ground truth was determined by the consensus of two expert listeners who made their determinations independently. To determine the ground truth, each listener was given 10 recordings of the test signal from the target implementation, and 10 recordings from the implementation under test. They were asked to make a determination regarding whether these recordings were made using implementations that were functionally equivalent, or whether there was a detectable difference. Each listener

Table 1: *Evaluation of feature representations and decision methods: Are the implementations equivalent? Bold indicates agreement.*

Preset Number	Human Listeners	Samples Vote %	Samples log KL	FFT Vote %	FFT log KL	STFT Vote %	STFT log KL	MFCC Vote %	MFCC log KL
3524	*no*	**10%**	**16.65**	**10%**	**25.95**	**10%**	**24.32**	**10%**	**16.25**
3526	*no*	**10%**	**16.94**	**10%**	**26.66**	**10%**	**24.62**	**10%**	**16.35**
615	*no*	**30%**	**13.44**	**10%**	**26.90**	**10%**	**22.27**	**0%**	**16.78**
7613	*no*	100%	**13.71**	**0%**	**18.78**	**10%**	**18.93**	**0%**	**10.62**
5012	*no*	100%	**12.10**	100%	13.65	100%	12.93	**0%**	9.47
4311	*no*	50%	**12.97**	**0%**	**15.39**	**0%**	**15.10**	**0%**	**11.65**
1411	*no*	**0%**	**15.60**	**0%**	**28.39**	**0%**	**22.82**	**0%**	**17.61**
5430	*no*	100%	**12.63**	**0%**	13.63	**0%**	**23.62**	**0%**	**14.67**
2211	*no*	**0%**	**15.24**	**0%**	**27.47**	**0%**	**22.02**	**0%**	**17.90**
221	*no*	100%	**12.13**	100%	**14.31**	90%	**13.45**	100%	9.97
225	*no*	100%	11.93	**0%**	**17.89**	100%	**13.04**	100%	9.58
4720	*yes*	**100%**	14.75	0%	**13.78**	0%	13.95	0%	12.72
5809	*yes*	**80%**	14.79	0%	**13.41**	10%	16.51	**90%**	**9.64**
224	*yes*	**50%**	**11.93**	40%	17.23	**60%**	13.99	**100%**	11.80
1413	*yes*	**50%**	**11.24**	**60%**	**12.56**	**100%**	**11.34**	**80%**	**9.59**
815	*yes*	**100%**	**11.87**	0%	**12.52**	**100%**	**11.76**	**100%**	**8.21**
1116	*yes*	0%	16.97	0%	19.61	0%	18.39	**100%**	11.63
4510	*yes*	**100%**	14.14	0%	**12.64**	**100%**	**11.80**	**100%**	**8.97**
12	*yes*	**100%**	**−4.79**	**100%**	**7.82**	**100%**	**2.26**	**100%**	**7.41**
Correct %		63.2%	73.7%	57.9%	78.9%	68.4%	73.7%	**84.2%**	68.4%

has been trained as a musician and recording engineer and should be considered an expert in the field of listening to musical effects. Neither listener was aware of the results from the other listener, or of the results of the QuAALM listening machine at the time of their determination.

The QuAALM listening machine was then run on the same set of recordings from each of the 19 effects and reported a Kullback-Leibler divergence information gain, and Voting Classifier percentage for each feature type. The results of this test are reported in Section six.

5.2. Experiment 2: Validation and Comparison against Human Listeners

Based on the results of Experiment 1, six listeners meeting the same qualifications as Experiment 1 were asked to make independent determinations of an additional 18 randomly chosen effects. These determinations were analyzed for consistency across the human listeners, and compared to the QuAALM machine results running using the Voting Classifier test and MFCC features. This analysis and results are included in Section six.

6. RESULTS

6.1. Experiment 1: Determination of Best Signal Representation and Decision Algorithm

The results of Experiment 1 are shown in Table 1. For each preset tested, the Vote % and log KL divergence are reported for each of the four feature types. The natural logarithm of the KL divergence is reported because the range of the KL divergence for such high dimensional data makes these values hard to display. After seeing the results, a threshold for the log KL divergence was chosen to optimize the results for each feature type. In a final implementation, the threshold would be considered a user parameter, and for this test an oracle threshold was chosen to evaluate this test on it's optimal possible performance. A reported log KL divergence below this threshold is scored as an equivalent implementation and a value above this threshold is scored as not equivalent. These thresholds are reported in Table 2.

Table 2: *Log KL divergence thresholds*

Samples	FFT	STFT	MFCC
12	14	13	10

Similarly, the Vote % represents the average of the votes for which the implementations were equivalent in the 10 runs of the voting classifier. A Vote % greater than or equal to 50% is scored as an equivalent implementation, below 50% is not. In Table 1, scores which agree with the judgment of the human listeners are in boldface while those that disagree are not. Finally, a percentage of presets for which QuAALM agrees with the human listeners is calculated for each test and reported at the bottom of Table 1.

From these results we can see that both the KL divergence and voting classifier methods have some success depending on the features used. However, the clear standout is the voting classifier method when used with MFCC features. Not only does it show the highest percentage of correct classifications, but for the instances where it passed, the classification margins are wider. This implies that there might be a higher robustness to noise using this test. Because of these results, we chose the voting classifier method operating on MFCC features for our deployment and run the validation tests in Experiment 2 using this method.

Table 3: *Validation and comparison against human listeners: Are the implementations equivalent?*

PresetNumber	Listener1	Listener2	Listener3	Listener4	Listener5	Listener6	Consensus	QuAALM	Verdict
234	yes	yes	yes	yes	yes	yes	yes	yes	*pass*
322	yes	yes	yes	yes	yes	yes	yes	yes	*pass*
536	yes	yes	yes	yes	yes	yes	yes	no	*fail*
1333	yes	yes	yes	no	yes	yes	yes	yes	*pass*
1910	yes	yes	yes	yes	yes	yes	yes	yes	*pass*
2317	no	no	no	no	no	no	no	no	*pass*
3211	yes	yes	yes	yes	yes	yes	yes	yes	*pass*
4034	yes	yes	yes	yes	yes	no	yes	yes	*pass*
4138	yes	yes	yes	yes	yes	yes	yes	yes	*pass*
4727	no	yes	no	yes	no	no	no	no	*pass*
4814	no	yes	no	yes	yes	yes	yes	no	*fail*
5729	yes	yes	yes	no	yes	yes	yes	yes	*pass*
5911	yes	yes	yes	yes	no	yes	yes	yes	*pass*
6663	yes	yes	yes	yes	yes	yes	yes	yes	*pass*
6910	no	no	no	no	no	no	no	no	*pass*
7211	yes	yes	yes	yes	yes	yes	yes	no	*fail*
7419	yes	no	no	no	no	yes	no	no	*pass*
8214	no	no	no	no	no	no	no	no	*pass*

Table 4: *Validation scores*

Precision	Recall	F_1
1.0	0.625	0.8696

6.2. Experiment 2: Validation and Comparison against Human Listeners

For Experiment 2, we randomly chose 18 presets from the H8000, making sure that no two were from the same bank, and tested them on both the H8000 and H9000 hardware. As in Experiment 1, we used a digital audio connection and tested at a 48 kHz sampling rate, the test signal consisted of a 3 second long Gaussian White Noise burst, 3 seconds of silence, a 3 second exponential sine chirp, and a final 3 seconds of silence. Ten independent recordings were made of both the H8000 and the H9000 for each preset.

The 20 recordings were each listened to by the six listeners, independently and without sharing their results, and a determination was made as to whether the two implementations sounded equivalent. For 11 of the 18 presets, all listeners agreed on their conclusions; 8 times that the implementations were the same, and 3 times that they were different. On a further 4, a single listener was in the minority, while the final 3 presets were a vote of 4 to 2. No presets came to a split decision among the 6 listeners. We believe that this shows the listeners have a high degree of consistency when determining if implementation differences were perceptually equivalent. These preset numbers and results are tabulated in Table 3.

The final ground truth consensus was reached by a vote of the human listeners and is also tabulated in Table 3.

For each preset, the same 20 recordings were fed into the QuAALM machine running the voting classifier method on MFCC features, and its judgment was recorded in Table 3, along with a comparison of its judgment and the ground truth consensus.

We can see that the QuAALM machine disagreed with the human listeners for only 3 of the 18 presets, and in each of these instances it reported a false negative rather than a false positive. This leads to a Precision score of 1, a Recall score of 0.625, and an F_1 score of 0.8696. These values are shown in Table 4.

As mentioned earlier, as a practical consideration, the perfect Precision score and relatively high F_1 score satisfied our goals and allowed us to put QuAALM into service testing real implementations.

7. CONCLUSIONS

In this paper, we developed an automated listening machine which meets the qualifications to serve as the first line of defense for quality assurance and automated testing of our effects hardware. In doing so we also developed a more robust framework for analyzing the similarity of effects utilizing some amount of random time variation. This framework may have applications to other listening intensive work like the automatic labeling of effect type.

The specific implementation described here was sufficient to serve our purposes, however, many improvements could be considered. Some potential improvements might come from exploring different test signals, or relaxing the strict independence criteria in the probability model, as we expect that there is likely some dependence between these samples. Additionally, given the limited test data reported here, we expect that the QuAALM machine might be failing for some specific types of effects. A larger data set will allow further investigation into these effect type and may lead us to new ways to improve the system.

Additionally, the comparisons made here were between two implementations of the same digital effects. However, the underlying technique should be useful to in determining the quality of analog models for certain types of randomized effects like chorus, flanger, phaser, or rotary speaker emulations. It would be interesting to see if this were the case.

8. ACKNOWLEDGMENTS

Thank you to Jeff Schumacher, Pete Bischoff, Patrick Flores, and Nick Solem for performing listening tests.

9. REFERENCES

[1] Shijun Xiang, Jiwu Huang, and Rui Yang, "Time-scale invariant audio watermarking based on the statistical features in time domain," in *International Workshop on Information Hiding*. Springer, 2006, pp. 93–108.

[2] Michael Arnold, "Audio watermarking: Features, applications and algorithms," in *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*. IEEE, 2000, vol. 2, pp. 1013–1016.

[3] Christian Neubauer and JÃ $\frac{1}{4}$rgen Herre, "Digital watermarking and its influence on audio quality," in *Audio Engineering Society Convention 105*, Sep 1998.

[4] Daniele Barchiesi and Joshua Reiss, "Reverse engineering of a mix," *J. Audio Eng. Soc*, vol. 58, no. 7/8, pp. 563–576, 2010.

[5] Stanislaw Gorlow and Joshua D Reiss, "Model-based inversion of dynamic range compression," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 7, pp. 1434–1444, 2013.

[6] Luwei Yang, Khalid Z Rajab, and Elaine Chew, "The filter diagonalisation method for music signal analysis: framewise vibrato detection and estimation," *Journal of Mathematics and Music*, pp. 1–19, 2017.

[7] Elias Pampalk, Simon Dixon, and Gerhard Widmer, "On the evaluation of perceptual similarity measures for music," in *of: Proceedings of the Sixth International Conference on Digital Audio Effects (DAFx-03)*, 2003, pp. 7–12.

[8] Marko Helén and Tuomas Virtanen, "A similarity measure for audio query by example based on perceptual coding and compression," in *Proc. 10th Int. Conf. Digital Audio Effects (DAFX)*, 2007.

[9] Tuomas Virtanen and Marko Helén, "Probabilistic model based similarity measures for audio query-by-example," in *Applications of Signal Processing to Audio and Acoustics, 2007 IEEE Workshop on*. IEEE, 2007, pp. 82–85.

[10] Mike Elliott, "How to Null Test Your Gear: Part 1," Available at https://music.tutsplus.com/tutorials/how-to-null-test-your-gear-part-1–cms-22425, accessed April 09, 2017.

[11] Md Sahidullah and Goutam Saha, "Design, analysis and experimental evaluation of block based transformation in mfcc computation for speaker recognition," *Speech Communication*, vol. 54, no. 4, pp. 543–565, 2012.

[12] Gordon Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Transactions on Information Theory*, vol. 14, no. 1, pp. 55–63, 1968.

[13] Yuichiro Anzai, *Pattern recognition and machine learning*, chapter Linear Models for Classification, pp. 200–203, Elsevier, 2012.

[14] Yuichiro Anzai, *Pattern recognition and machine learning*, chapter Mixture Models and EM, pp. 450–451, Elsevier, 2012.

[15] Giovanni Seni and John F Elder, "Ensemble methods in data mining: improving accuracy through combining predictions," *Synthesis Lectures on Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 1–126, 2010.

[16] D. M. W. Powers, "Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.