applied sciences

MDPI

Article

Semantically Controlled Adaptive Equalisation in Reduced Dimensionality Parameter Space †

Spyridon Stasis *, Ryan Stables * and Jason Hockman *

Digital Media Technology Lab, Birmingham City University, Birmimgham B42 2SU, UK

* Correspondence: spyridon.stasis@bcu.ac.uk (S.S.); ryan.stables@bcu.ac.uk (R.S.); jason.hockman@bcu.ac.uk (J.H.); Tel.: +4412-1331-7957 (R.S.); +4412-1202-2386 (J.H.)

† This paper is an extended version of our paper published in the 18th International Conference on Digital Audio Effects, Trondheim, Norway, 30 November–3 December 2015.

Academic Editor: Vesa Valimaki

Received: 24 February 2016; Accepted: 5 April 2016; Published: 20 April 2016

Abstract: Equalisation is one of the most commonly-used tools in sound production, allowing users to control the gains of different frequency components in an audio signal. In this paper we present a model for mapping a set of equalisation parameters to a reduced dimensionality space. The purpose of this approach is to allow a user to interact with the system in an intuitive way through both the reduction of the number of parameters and the elimination of technical knowledge required to creatively equalise the input audio. The proposed model represents 13 equaliser parameters on a two-dimensional plane, which is trained with data extracted from a semantic equalisation plug-in, using the timbral adjectives *warm* and *bright*. We also include a parameter weighting stage in order to scale the input parameters to spectral features of the audio signal, making the system adaptive. To maximise the efficacy of the model, we evaluate a variety of dimensionality reduction and regression techniques, assessing the performance of both parameter reconstruction and structural preservation in low-dimensional space. After selecting an appropriate model based on the evaluation criteria, we conclude by subjectively evaluating the system using listening tests.

Keywords: equalisation; adaptive audio effects; semantics; dimensionality reduction; intelligent music production

1. Introduction

Equalisation, as described in [1], is an integral part of the music production workflow, with applications in live sound engineering, recording, music production, and mastering, in which multiple frequency dependent gains are imposed upon an audio signal. Generally, the process of equalisation can be categorised under one of the following headings as described in [2], corrective equalisation: in which problematic frequencies are often attenuated in order to prevent issues such as acoustic feedback, and creative equalisation: in which the audio spectrum is modified to achieve a desired timbral aesthetic. Whilst the former is primarily based on adapting the effect parameters to the changes in the audio signal, the latter often involves a process of translation between a perceived timbral adjective such as *bright*, *flat*, or *sibilant* and an audio effect input space, by which a music producer must reappropriate a perceptual representation of a timbral transformation as a configuration of multiple parameters in an audio processing module. As music production is an inherently technical process, this mapping procedure is not necessarily trivial, and is made more complex by the source-dependent nature of the task.

2. Background

2.1. Semantically-Controlled Audio Effects

Engineers and producers generally use a wide variety of timbral adjectives to describe sound, each with varying levels of agreement. By modelling these adjectives, we are able to provide perceptually meaningful abstractions, which lead to a deeper understanding of musical timbre and systems that facilitate the process of audio manipulation. The extent to which timbral adjectives can be accurately modelled is defined by the level of exhibited agreement, a concept investigated in [3], in which terms such as *bright*, *resonant*, and *harsh* all exhibit strong agreement scores, and terms such as *open*, *hard*, and *heavy* all show low subjective agreement scores. It is common for timbral descriptors to be represented in low-dimensional space; *brightness*, for example, is shown to exhibit a strong correlation with spectral centroid [4,5] and has further dependency on the fundamental frequency of the signal [6]. Similarly, studies such as [7,8] demonstrate the ability to reduce complex data to lower-dimensional spaces using dimensionality reduction.

Recent studies have also focused on modification of the audio signal using specific timbral adjectives, where techniques such as spectral morphing [9] and additive synthesis [10] have been applied. For the purposes of equalisation, timbral modification has also been implemented via psychoacoustic measurements such as loudness [11], spectral masking [12], and semantically-meaningful controls and intuitive parameter spaces. SocialEQ [13], for example, collects timbral adjective data via a web interface and approximates the configuration of a graphic equaliser curve using multiple linear regression. Similarly, subjEQt [14] provides a two-dimensional interface, created using a Self-Organising Map, in which users can navigate between presets such as *boomy*, *warm*, and *edgy* using natural neighbour interpolation. This is a similar model to 2DEQ [15], in which timbral descriptors are projected onto a two-dimensional space using Principal Component Analysis (PCA). The Semantic Audio Feature Extraction (SAFE) project provides a similar non-parametric interface for semantically controlling a suite of audio plug-ins, in which semantics data is collected within a given Digital Audio Workstation (DAW). Adaptive presets can then be selectively derived based on audio features, parameter data, and music production metadata.

2.2. Aims

In this study, we propose a system that projects the controls of a parametric equaliser comprising five biquad filters, as detailed in [16], arranged in series onto an editable two-dimensional space, allowing the user to manipulate the timbre of an audio signal using an intuitive interface. Whilst the axes of the two-dimensional space are somewhat arbitrary, underlying timbral characteristics are projected onto the space via a training stage using two-term musical semantics data. In addition to this, we propose a signal processing method of adapting the parameter modulation process to the incoming audio data based on feature extraction applied to the long-term average spectrum (LTAS), as detailed in [17–19], capable of running in near-real-time. The model is implemented using the SAFE architecture (detailed in [20]), and is provided as an extension of the current Semantic Audio Parametric Equaliser (available for download at [21]), as shown in Figure 1a.

3. Methodology

In order to model the desired relationship between the two parameter spaces, a number of problems must be addressed. Firstly, the data reduction process should account for maximal variance in high-dimensional space without bias towards a smaller subset of the equaliser parameters. Similarly, we should be able to map to the high-dimensional space with minimal reconstruction error, given a new set of (x, y) coordinates. This process of mapping between spaces is nontrivial, due to loss of information in the reconstruction process. Furthermore, the low-dimensional parameter space should be configured in a way that preserves an underlying timbral characteristic in the data,

thus allowing a user to transform the incoming audio signal in a musically meaningful way. Finally, the process of parameter space modification should not be agnostic of the incoming audio signal, meaning that any mapping between the two-dimensional plane and the equaliser parameters should be expressed as a function of the (x, y) coordinates and some representation of the signal spectral energy. In addition to this, the system should be capable of running in near-real time, enabling its use in a DAW environment.

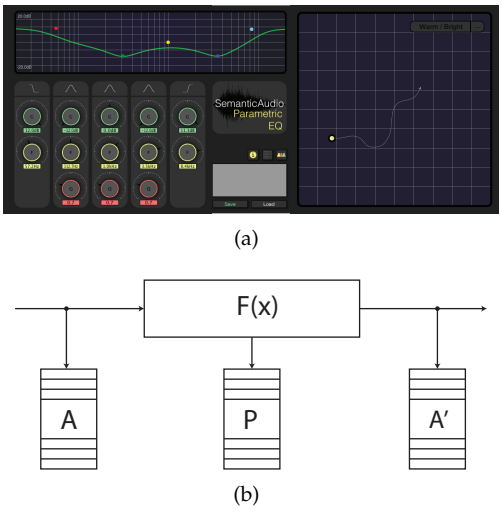


Figure 1. An overview of the Semantic Audio Feature Extraction (SAFE) equaliser and its feature extraction architecture. (**a**) The extended Semantic Audio Equalisation plug-in with the two-dimensional interface. To modify the *brightness*/*warmth* of an audio signal, a point is positioned in two-dimensional space; (**b**) The SAFE feature extraction process, where A represents the audio features captured before the effect is applied, A' represents the features captured after the effect is applied, and P represents the parameter vector.

To address these problems, we develop a model that consists of two phases. The first is a training phase, in which a map is derived from a corpus of semantically-labelled parameter data, and the second is an implementation phase in which a user can present (x, y) coordinates and an audio spectrum, resulting in a 13-dimensional vector of parameter state variables. To optimise the mapping process, we experiment with a combination of 6 dimensionality reduction techniques and 5 reconstruction methods, followed by a stacked-autoencoder (sAE) model that encapsulates both the dimensionality reduction and reconstruction processes. The techniques were chosen to represent a variable range of complexity and nonlinearity, and were intended to provide a selection of possible solutions to the problem, in which the highest performing section would be used for implementation. With the intention of scaling the parameters to the incoming audio signal, we derive a series of weights based on a selection of features, extracted from the signal LTAS coefficients. To evaluate the model performance under a range of conditions, we train it with binary musical semantics data and measure both objective and subjective performance based on the reconstruction of the input space and the structural preservation in reduced dimensionality space.

3.1. Dataset

For the training of the model, we compile a dataset of 800 semantically-annotated equaliser parameter space settings, comprising 40 participants equalising 10 musical instrument samples using two descriptive terms: *warm* and *bright*. To do this, participants were presented with the musical instrument samples in a DAW and asked to use a parametric equaliser to achieve the two timbral settings. After each setting was recorded, the data were recorded and the equaliser was reset to unity gain. During the test, samples were presented to the participants in a random order across separate

DAW channels. Furthermore, the musical instrument samples were all performed unaccompanied, were Root Mean Square (RMS) normalised and ranged from 20 to 30 s in length. All of the participants had normal hearing, were aged 18–40, and all had at least 3 years' music production experience.

The descriptive terms (*warm* and *bright*) were selected for a number of reasons; firstly, the agreement levels exhibited by participants tend to be high (as suggested by [3]), meaning there should be less intra-class variance when subjectively assigning parameter settings. When measured using an agreement metric, defined by [13] as the log number of terms over the trace of the covariance matrix, *warm* and *bright* were the two highest ranked terms in a dataset of 210 unique adjectives. Secondly, the two terms are deemed to be sufficiently different enough to form an audible timbral variation in low dimensional space. While the two terms do not necessarily exhibit orthogonality (for example, *brightness* can be modified with constant *warmth* [9]), they have relatively dissimilar timbral profiles, with *brightness* widely accepted to be highly correlated with the signal's spectral centroid, and *warmth* often attributed to the ratio of the first three harmonics to the remaining harmonic partials in the magnitude spectrum [22].

The parameter settings were collected using a modified build of the SAFE data collection architecture, in which descriptive terms, audio feature data, parameter data, and metadata can be collected remotely within the DAW environment and uploaded to a server. As illustrated in Figure 1b, the SAFE architecture allows for the capture of audio feature data before and after processing has been applied. Similarly, the interface parameters P are captured and stored in a linked database. For the purpose of this experiment, the architecture was modified by adding the functionality to capture LTAS coefficients, with a window size of 1024 samples and a hop size of 256.

While the SAFE project comprises a number of DAW plug-ins, we focus solely on the parametric equaliser, which utilises five biquad filters arranged in series, consisting of a low-shelving filter (LS), three peaking filters (Pf_n), and a high-shelving filter (HS), where the LS and HS filters each have two parameters and the (Pf_n) filters each have three, as described in Table 1.

Table 1. A list of the parameter space variables and their ranges of possible values, taken from the Semantic Audio Feature Extraction (SAFE) parametric equaliser interface.

n	Assignment	Range	*n*	Assignment	Range
0	LS gain	−12–12 dB	7	Pf_1 Q	0.1–10 Hz
1	LS Freq	22–1000 Hz	8	Pf_2 Gain	−12–12 dB
2	Pf_0 Gain	−12–12 dB	9	Pf_2 Freq	220–10,000 Hz
3	Pf_0 Freq	82–3900 Hz	10	Pf_2 Q	0.1–10 Hz
4	Pf_0 Q	0.1–10 Hz	11	HS Gain	−12–12 dB
5	Pf_1 Gain	−12–12 dB	12	HS Freq	580–20,000 Hz
6	Pf_1 Freq	180–4700 Hz			

3.2. Evaluation Criteria

To evaluate the model under various conditions and to select an appropriate mapping topology, we apply objective metrics to the data during the dimensionality reduction and reconstruction processes. These allow us to evaluate the extent to which (1) the dimensionality reduction technique retains the structure of the high-dimensional data (*trustworthiness, continuity, K-Nearest Neighbours (K-NN)*), (2) the classes are separable in low-dimensional space (*Jeffries–Matusita Distance*), and (3) the system accurately reconstructs the high-dimensional parameter space (*reconstruction error*).

3.2.1. Trustworthiness and Continuity

To evaluate the structural preservation of each technique, the metrics *trustworthiness* and *continuity* [23] are applied to the dataset. Here, the distance of point i in high-dimensional space is measured against its k closest neighbours using rank order, and the extent to which each rank changes in low-dimensional space is measured. For n samples, let $r(i, j)$ be the rank in distance of

sample i to sample j in the high-dimensional space U_i^k. Similarly, let $\hat{r}(i,j)$ be the rank of the distance between sample i and sample j in low-dimensional space V_i^k. Using the k-nearest neighbours, the map is considered *trustworthy* if these k neighbours are also placed close to point i in the low-dimensional space, as shown in Equation (1).

$$T(k) = 1 - \frac{2}{nk(2n - 3k - 1)} \sum_{i=1}^{n} \sum_{j \in U_i^{(k)}} (r(i,j) - k) \tag{1}$$

Similarly, *continuity* (shown in Equation 2) measures the extent to which original clusters of datapoints are preserved, and can be considered the inverse to *trustworthiness*, finding sample points that are close to point i in low-dimensional space, but not in the high-dimensional plane.

$$C(k) = 1 - \frac{2}{nk(2n - 3k - 1)} \sum_{i=1}^{n} \sum_{j \in V_i^{(k)}} (\hat{r}(i,j) - k) \tag{2}$$

In both of these equations, a normalising factor is used to bound the *trustworthiness* and *continuity* scores between 0 and 1. These measures evaluate the extent to which the local structure of the original dataset is preserved in a low-dimensional map; this is described in [24], where it is shown that the local structure of the dataset needs to be retained for a successful map of the datapoints.

3.2.2. K-NN

In order to measure the similarities in inter-class structures within the high and low dimensional space, we apply a K-NN classifier with $k = 1$, as described in [25], and then measure the differences in classification accuracies. The nearest neighbours are found using Euclidean distances with 13 and 2 dimensions, respectively. The accuracies are derived using K-fold cross validation with K = 20, where 20% of the data is partitioned for testing. This allows us to measure the extent to which the between-class structures have been preserved in the reduction process, and effectively acts as a supervised structural preservation metric.

3.2.3. Jeffries–Matusita Distance

In order to evaluate the extent to which timbral descriptors lie at opposing ends of the mapped parameter space, we can measure the extent to which the timbre classes are separable using a distance metric. Typically, this can be done by finding the divergence between class distributions using a technique such as Kullback–Leibler Divergence (KLD), as we proposed in [26]; however, as explained in [27], this does not satisfy the triangle inequality based on the measurement's asymmetry. While two-sided KLD addresses this, as explained in [28], [29] proposes Jeffries–Matusita Distance (JMD) as a more appropriate alternative. JMD (as shown in Equation 4) is a metric derived from the Bhattacharya (BH) distance, as in Equation (3), which bounds the output of the measure from 0 (no separability) to 2 (perfect separability).

$$BH_{i,j} = \frac{1}{8}(m_i - m_j)^T \left(\frac{S_i + S_j}{2}\right)^{-1} (m_i - m_j) + 0.5 \ln\left(\frac{0.5(|S_i + S_j|)}{\sqrt{|S_i|\,|S_j|}}\right) \tag{3}$$

$$JMD_{1,2} = \sqrt{2(1 - e^{-BH_{i,j}})} \tag{4}$$

Here m represents the mean and S represents the covariance of classes i and j, respectively.

3.2.4. Reconstruction Error

To measure the reconstruction accuracy (low-to-high-dimensionality mapping) of the model, we measure the input/output error for each pair-wise combination of dimensionality reduction and reconstruction techniques by computing the mean absolute error between predicted and actual

parameter values. This is done using K-fold cross validation with $k = 20$ iterations, and a test partition size of 20% (160 training examples). As some of the dimensionality reduction techniques are unable to embed new information into the reduced-dimensionality space, the first part of the test process (*i.e.*, the prediction of new low-dimensional values as implemented in [26]) was omitted, and only regression and interpolation techniques were evaluated.

3.3. Subjective Evaluation

Using the metrics defined in Section 3.2, we are able to select an appropriate model which is capable of accurately reducing the dataset while preserving the data structure and accurately reconstructing the input parameters with minimal error. To validate this, we implement subjective user tests in which participants are asked to equalise a series of audio samples using the reduced-dimensionality interface. To do this, 10 participants were asked to apply the process to 10 input sounds using only the two-dimensional interface. Each participant was asked to achieve a *warm* or *bright* output sound for each stimuli. During the test, samples were presented to participants in a random order across separate DAW channels, and the equaliser parameters remained hidden. No indication was given as to the underlying distribution of datapoints. The stimuli comprised unaccompanied musical instrument samples and ranged from 20 to 30 s in length. The samples were primarily taken from electric guitars and included a variety of genres, taken from the Mixing Secrets Multitrack Audio Dataset [30]. All of the participants had normal hearing, were aged 18–35, and had varied music production experience, from 0 to 5 years.

4. Model

The proposed system maps between the equaliser parameter space, consisting of 13 filter parameters and a two-dimensional plane, while preserving the context-dependent nature of the audio effect. After an initial training phase, the user can then submit (x, y) coordinates to the system using a track-pad interface, resulting in a timbral modification via the corresponding filter parameters. To demonstrate this, we train the model with two class (*bright*, *warm*) musical semantics data taken from the SAFE equaliser database, thus resulting in an underlying transition between opposing timbral descriptors in two-dimensional space. By training the model in this manner, we intend to retain the high-dimensional structure of the dataset in the two-dimensional space while minimising the reconstruction error inherent to dimensionality reduction methods.

The model (illustrated in Figure 2) has two key operations. The first involves weighting the parameters by computing the vector $\alpha_n(A)$ from the input signal long-term spectral energy (A). We can then modify the parameter vector (P) to obtain a weighted vector (P'). The second component scales the dimensionality of (P'), resulting in a compact audio-dependent representation. During the model implementation phase, we apply an unweighting procedure based on the (x, y) coordinates and the signal modified spectrum. This is done by multiplying the estimated parameters with the inverse weight vector, resulting in an approximation of the original parameters. In addition to the weighting and dimensionality reduction stages, a scale-normalisation procedure is applied, aiming to convert the ranges of each parameter (given in Table 1), to $(0 < p_n < 1)$. This converts the data into a suitable format for dimensionality reduction.

4.1. Parameter Scaling

As the configuration of the filter parameters assigned to each descriptor by the user during equalisation is likely to vary based on the audio signal being processed, the first requirement of the model is to apply weights to the parameters based on knowledge of the audio data at the time of processing. To do this, we selectively extract features from the signal LTAS before and after the filter is applied. This is possible due to the configuration of the data collection architecture, highlighted in Figure 1b. The weights (α_m) can then be expressed as a function of the LTAS, where the function's definition varies based on the parameter representation (*i.e.*, gain, centre frequency, or bandwidth of

the corresponding filter). We use the LTAS to prevent the parameters from adapting each time a new frame is read. In practice, we are able to do this by presenting users with means to store the audio data, rather than continually extracting it from the audio stream. Each weighting is defined as the ratio between a spectral feature taken from the filtered audio signal (A'_k) and the signal filtered by an enclosing rectangular window (R_k). Here, the rectangular window is bounded by the minimum and maximum frequency values attainable by the observed filter $f_k(A)$.

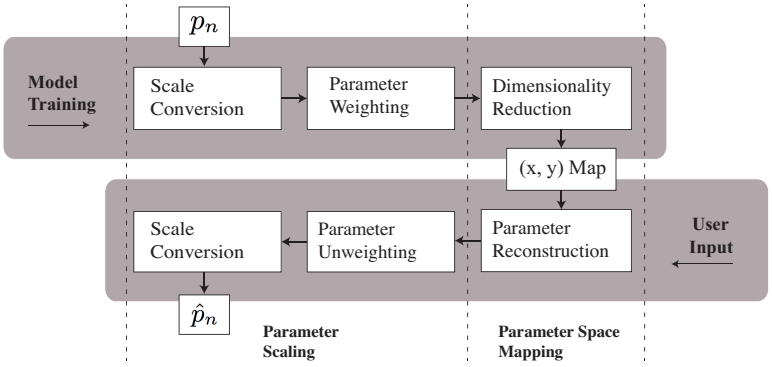


Figure 2. An overview of the proposed model. The grey horizontal paths represent training and implementation (user input) phases.

We can define the equaliser as an array of biquad functions arranged in series, as depicted in Equation (5).

$$\begin{gathered} f_k = f_{k-1}(A, \vec{P}_{k-1}) \\ k = 1, \ldots, K-1 \end{gathered} \tag{5}$$

Here, $K = 5$ represents the number of filters used by the equaliser and f_k represents the k^{th} biquad function, which we can define by its transfer function, given in Equation (6).

$$H_k(z) = c \cdot \frac{1 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{6}$$

The LTAS is then modified by the filter as in Equation (7) and the weighted parameter vector can be derived using the function expressed in Equation (8).

$$A'_k = |H_k(e^{j\omega})| A_k \tag{7}$$

$$p'_n = \alpha_m(k) \cdot p_n \tag{8}$$

where p_n is the n^{th} parameter in the vector P. The weighting function is then defined by the parameter type (m), where $m = 0$ represents gain, $m = 1$ represents centre-frequency, and $m = 2$ represents bandwidth. For gain parameters, the weights are expressed as a ratio of the spectral energy in the filtered spectrum (A') to the spectral energy in the enclosing rectangular window (R_k), derived in Equation (9) and illustrated in Figure 3.

$$\alpha_0(k) = \frac{\sum_i (A'_k)_i}{\sum_i (R_k)_i} \tag{9}$$

For frequency parameters ($m = 1$), the weights are expressed as a ratio of the respective spectral centroids of A' and R_k, as demonstrated in Equation (10), where bn_i are the corresponding frequency bins.

$$\alpha_1(k) = \left(\frac{\sum_i (A'_k)_i bn_i}{\sum_i (A'_k)_i} \right) / \left(\frac{\sum_i (R_k)_i bn_i}{\sum_i (R_k)_i} \right) \tag{10}$$

Finally, the weights for bandwidth parameters ($m = 2$) are defined as the ratio of spectral spread exhibited by both A' and R_k. This is demonstrated in Equation (11), where $(x)_{sc}$ represents the spectral centroid of x.

$$\alpha_2(k) = \left(\frac{\sum_i \left(bn_i - (A'_k)_{sc}\right)^2 (A'_k)_i}{\sum_i (A'_k)_i} \right) / \left(\frac{\sum_i \left(bn_i - (R_k)_{sc}\right)^2 (R_k)_i}{\sum_i (R_k)_i} \right) \tag{11}$$

During the implementation phase, retrieval of the unweighted parameters, given a weighted vector, can be achieved by simply multiplying the weighted parameters with the inverse weights vector, as in Equation (12).

$$\hat{p}_n = \alpha_m^{-1}(k) \cdot p'_n \tag{12}$$

where $\hat{p}$ is a reconstructed version of p, after dimensionality reduction has been applied.

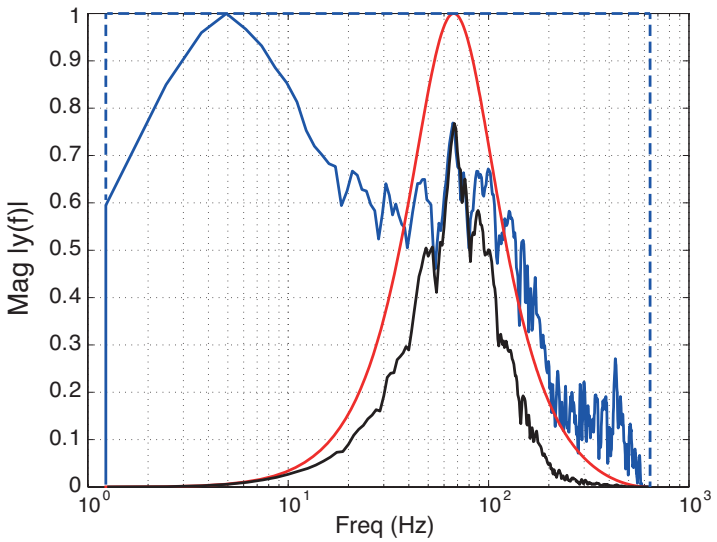


Figure 3. An example spectrum taken from an input example, weighted by the biquad coefficients, where the red line represents a peaking filter, the black line represents the biquad-filtered spectrum, and the blue line represents the spectral energy in the rectangular window (R_k).

To ensure the parameters are in a consistent format for each of the dimensionality scaling algorithms, a scale normalisation procedure is applied using Equation (13), where during the training process, the p_{min} and p_{max} represent the minimum and maximum values for each parameter (given in Table 1), and q_{min} and q_{max} represent 0 and 1. During the implementation process, these values are exchanged such that q_{min} and q_{max} represent the minimum and maximum values for each parameter and p_{min} and p_{max} represent 0 and 1.

$$\rho_n = \frac{(p_n - q_{min})(p_{max} - p_{min})}{q_{max} - q_{min}} + p_{min} \tag{13}$$

Additionally, a sorting algorithm was used to place the three mid-band filters in ascending order based on their centre frequency. This prevents normalisation errors due to the frequency ranges, allowing filters to be rearranged by the user.

4.2. Parameter Space Mapping

Once the filters have been weighted by the audio signal, the mapping from 13 equaliser variables to a two-dimensional subspace can be accomplished using a range of dimensionality reduction techniques. In this study, we expand on [26] and evaluate the performance of six dimensionality

reduction techniques. Here, the algorithms that were used for the dimensionality reduction are available as part of the dimensionality reduction toolbox in [31]. In addition to this, parameter space mapping is evaluated by measuring the quality of reduction with rank-based measures and nearest neighbour classification algorithms. In dimensionality reduction, the reconstruction process is often less common due to the nature of the task (e.g., feature optimisation, data reduction). We evaluate the efficacy of two regression-based techniques and three interpolation techniques at mapping two-dimensional interface variables to a vector of equaliser parameters. This is done by approximating functions using the weighted parameter data and measuring the reconstruction error. Finally, we evaluate an sAE model of data reduction, in which the parameter space is both reduced and reconstructed in the same algorithm; we are then able to isolate the reconstruction (decoder) stage for the implementation process.

Dimensionality reduction is implemented using the following techniques: PCA, a widely used method of embedding data into a linear subspace of reduced dimensionality by finding the eigenvectors of the covariance matrix, originally proposed by [32]; *Kernel PCA* (kPCA), a non-linear manifold mapping technique in which the eigenvectors are computed from a kernel matrix as opposed to the covariance matrix, as defined by [33]; *probabilistic PCA* (pPCA), a method that considers standard PCA as a latent variable model and makes use of an Expectation Maximisation (EM) algorithm, a method for finding the maximum-likelihood estimate of the parameters in an underlying distribution from a given data set, depending on unobserved latent variables [34] as described in [35]; *Factor Analysis* (FA), a statistical analysis technique that identifies the relationship between different variables of a dataset and groups those variables by the correlation of the underlying factors [36]; *Diffusion Maps* (DM), a technique inspired by the field of dynamical systems, reducing the dimensionality of data by embedding the original dataset in a low-dimensional space by retrieving the eigenvectors of Markov random walks [37]; *Linear Discriminant Analysis* (LDA), a supervised projection technique that maps to a linear subspace while maximising the separability between data points that belong to different classes (see [38]). As LDA projects the data-points onto the dimensions that maximise inter-class variance for C classes, the dimensionality of the subspace is set to $C - 1$. This means that in a binary classification problem such as ours, we need to reconstruct the second dimension arbitrarily. For each of the other algorithms, we select the first two variables for mapping, and for the kPCA algorithm, the feature distances are computed using a Gaussian kernel.

The parameter reconstruction process was implemented using the following techniques: *Linear Regression* (LR), a process by which a linear function is used to estimate latent variables; *Natural Neighbour Interpolation* (NaNI), a method for interpolating between scattered data points using Voronoi tessellation, as used by [14] for a similar application; *Nearest Neighbour Interpolation* (NeNI), an interpolation method where the query point takes the value of the nearest neighbour [39]; *Linear Interpolation* (LI), an interpolation technique that assumes a linear relationship between the existing points in a dataset; *Support Vector Regression* (SVR), a non-linear kernel-based regression technique (see [40]), for which we choose a Gaussian kernel function.

An autoencoder is an Artificial Neural Network (ANN) with a topology capable of learning a compact representation of a dataset by optimising a matrix of weights, such that a loss function representing the difference between the output and input vectors is minimised. Autoencoders can then be stacked together using the output of the prior layer as the input for the next in order to construct a deep network architecture. Each autoencoder is then trained individually, learning to minimise the reconstruction error between its input and the predicted output. This approach has been used for data compression [41], and by extension, dimensionality reduction. This type of ANN is often used in order to improve the classification accuracy of logistic regression [42]; however, since our problem involves data reconstruction as opposed to classification, a logistic layer is not implemented.

Network Topology

The autoencoder was built using the Theano Python library [43], where we observed an error of 0.086 using a single hidden layer with N (in this case $N = 2$) units. To reduce the error, a mirrored $[13-9-2]$ architecture was selected empirically, resulting in an error measurement of 0.08. To improve reconstruction accuracy further, noise was introduced at each stage in the network, as demonstrated by [44]. Here, the first autoencoder was corrupted with 0.6 magnitude noise, and the second with 0.5. This approach is able to further reduce the reconstruction error to 0.0784. Additionally, we replace the previously-used stochastic gradient descent algorithm with an *RMSprop* method [45] with a batch size of 10 as the pre-training and fine-tuning methods of optimization, and a learning rate of 0.01 and 0.001, respectively. This approach allows for faster optimization, as shown in [46]. For the weighted parameters, we found that a three-layer denoising autoencoder with an architecture of $[13-9-6-2]$ and noise of magnitude $(0.5, 0.4, 0.3)$ is able to outperform our two-layer denoising autoencoder model.

5. Results

5.1. Parameter Space Evaluation

To evaluate the extent to which structures in the parameter space are preserved in the reduced dimensionality map, we report the *trustworthiness*, *continuity*, and class-wise similarity (k-NN). This is applied to data shown in Figure 4a–g, in which a two-dimensional projection of the 13 equaliser parameters is given for both *warm* and *bright* samples in the dataset.

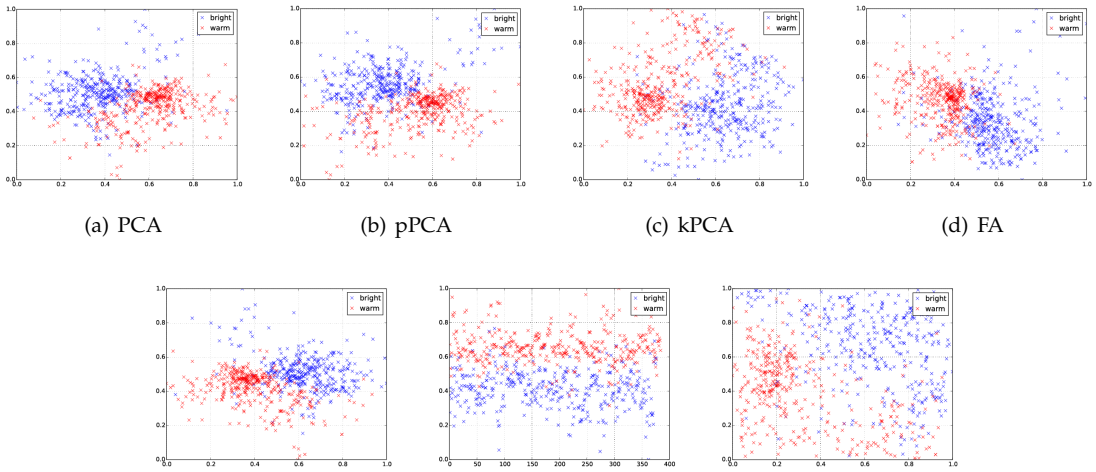
5.1.1. Low-Dimensional Mapping Accuracy

From Table 2 we show that for *trustworthiness*, pPCA achieves the highest rating (0.8426), with the sAE also performing similarly (0.842). The rest of the techniques are also able to achieve a high score, ranging from 0.81 for kPCA to 0.839 for standard PCA. The only technique that does not perform to the same standard is LDA, as the algorithm maximizes the separability of classes in the data instead of preserving the structure of the original dataset unrelated to its classes. For *continuity*, we can see that the majority of the techniques perform similarly, with scores ranging from 0.943 for the sAE to 0.958 for kPCA. However, as was the case with *trustworthiness*, LDA does not perform as well (0.868), due to the map reduction process.

Table 2. *Trustworthiness* and *continuity* scores for the different dimensionality reduction techniques (higher values are better) and classification accuracy of 1-NN classification

Technique	Trustworthiness	Continuity	1-NN Classification
Original	-	-	91.21%
PCA	0.8398	0.9541	87.61%
pPCA	0.8426	0.9567	87.92%
kPCA	0.8102	0.9583	86.14%
FA	0.8337	0.9490	86.19%
DM	0.8395	0.9533	87.89%
LDA	0.7292	0.8684	85.40%
sAE	0.8420	0.9439	84.01%

Trustworthiness and *continuity* metrics were used with a varying number of neighbours, ranging from 1 to 250. Here, the sAE exhibits higher scores for a lower number of neighbours (<120), as shown in Figure 5a—a result that suggests the system is better at retaining the local structure of the data, which is a necessary goal for a successful mapping technique. Furthermore, while the *continuity* score of the autoencoder is lower than the remaining dimensionality reduction techniques (Table 2), its error from the best performing technique in terms of *continuity* (kPCA) is only 0.015, which is deemed negligible.



(a) PCA (b) pPCA (c) kPCA (d) FA

(e) DM (f) LDA (g) sAE

Figure 4. Two-dimensional parameter-space representations using seven data reduction techniques, where the red data points are taken from parameter spaces described as *bright* and the blue points are described as *warm*. (**a**), (**c**), (**f**), and (**g**). PCA: Principal Components Analysis; pPCA: Probabilistic PCA; kPCA: Kernel PCA; FA: Factor Analysis; DM: Diffusion Maps; LDA: Linear Discriminant Analysis; sAE: stacked-autoencoder.

5.1.2. Class Preservation

The classification of 1-NN in the original dataset achieves an average of 91.21% for 100 iterations of the algorithm. None of the dimensionality reduction techniques are able to replicate this response, with pPCA achieving the highest score (87.92%), as seen in Table 2. On the other hand, the sAE achieved an accuracy of 84.01%, the lowest among the techniques being tested, 7.2% worse than the classification accuracy of the algorithm in the high-dimensional dataset. This result reveals that sAE is not as capable as other reduction techniques in preserving the classes on the low-dimensional space; however, as sAE is able to achieve better results than the other techniques for *trustworthiness* for a lower number of neighbours, and its performance in *1*-NN is not drastically worse (3.91%) than the best technique in pPCA, it can be considered a minor problem.

5.1.3. Class Separation

By applying JMD (Equation 4) to the dimensionality reduction techniques, we find that kPCA outperforms the rest of the techniques used, achieving 0.607, whereas the optimised autoencoder model performs slightly less favourably with a score 0.558, as shown in Table 3. The only technique that was excluded from this process was LDA, for two reasons: (1) it is a supervised technique that specifically maximizes the separability between the different classes in the low-dimensional space, and (2) in the context of our study, LDA has reduced the dataset to a single dimension, while all the other techniques have reduced the dimensionality to two dimensions. While class-separability is not necessarily correlated with accurate preservation of structure, high separability will allow users to effectively modulate between contrasting timbral descriptors.

Table 3. Jeffries–Matusita Distance (JMD) scores showing separation across different dimensionality reduction techniques.

Separability Measure	PCA	pPCA	kPCA	FA	DM	sAE
JMD	0.5142	0.5152	0.6076	0.4862	0.5125	0.5581

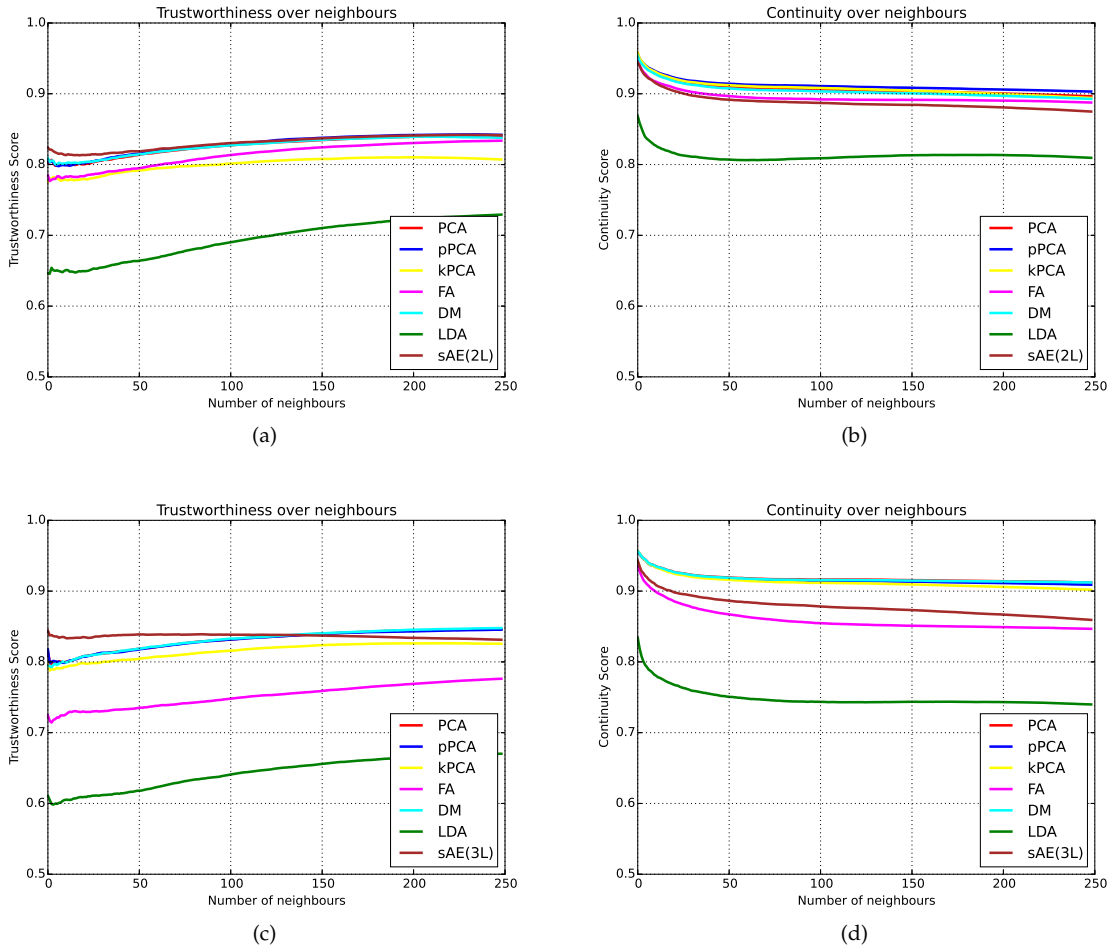


Figure 5. *Trustworthiness* and *continuity* plots across the different dimensionality reduction techniques for number of neighbors $(1:250)$. (**a**) Trustworthiness; (**b**) Continuity; (**c**) Trustworthiness (Weighted Parameters); (**d**) Continuity (Weighted Parameters).

5.2. Parameter Reconstruction Error

In [26], the sAE was able to achieve the lowest reconstruction error, 0.086, while the technique that came the closest to its accuracy was kPCA with support vector regression, achieving an error of 0.09. The sAE technique still outperforms all the other combinations of techniques, as can be seen in Table 4, achieving an overall error 0.074. It should also be noted that the sAE is able to reconstruct the most parameters of the equaliser (6) more accurately than any other combination of techniques.

5.3. Parameter Weighting

In order to evaluate the effectiveness of the signal specific weights, we measure the reconstruction accuracy of each system after the weights have been applied (see Table 5). Overall, the systems exhibit a general improvement in the reconstruction accuracy of the gain and Q parameters. All the systems have improved accuracy measurements, with the highest performing pair being PCA with SVR, achieving an error of 0.059. Similarly, the sAE with the same architecture, with hidden layer sizes $[9, 2]$, is able to achieve a reconstruction accuracy of 0.06—a further improvement from the 0.0748 error observed with unweighted parameters. For the weighted parameters we found that a three-layer denoising autoencoder was able to outperform our two-layer autoencoder, improving the reconstruction accuracy by 0.02.

Table 4. Mean reconstruction error per parameter using combinations of dimensionality reduction and reconstruction techniques, with the lowest reconstruction error highlighted in grey. The final column shows the mean (μ) error across all techniques, while the model with the lowest mean reconstruction error (Stacked Autoencoder, sAE) is highlighted in green. LR: Linear Regression; SVR: Support Vector Regression; NaNI: Natural Neighbour Interpolation; NeNI: Nearest Neighbour Interpolation; LI: Linear Interpolation.

P:	0	1	2	3	4	5	6	7	8	9	10	11	12	μ
PCA-LR	0.099	0.070	0.142	0.047	0.041	0.139	0.079	0.028	0.124	0.090	0.029	0.102	0.109	0.084
LDA-LR	0.194	0.070	0.150	0.047	0.041	0.171	0.082	0.028	0.116	0.090	0.030	0.123	0.106	0.096
kPCA-LR	0.081	0.070	0.136	0.047	0.040	0.150	0.082	0.027	0.130	0.084	0.029	0.120	0.107	0.085
pPCA-LR	0.099	0.069	0.138	0.046	0.039	0.142	0.078	0.027	0.126	0.092	0.030	0.104	0.108	0.084
DM-LR	0.104	0.070	0.138	0.047	0.040	0.139	0.081	0.027	0.126	0.091	0.031	0.102	0.106	0.085
FA-LR	0.151	0.068	0.156	0.042	0.040	0.143	0.068	0.029	0.144	0.084	0.030	0.103	0.094	0.089
PCA-SVR	0.086	0.064	0.123	0.046	0.040	0.137	0.079	0.028	0.125	0.089	0.031	0.097	0.095	0.080
LDA-SVR	0.196	0.068	0.152	0.048	0.040	0.171	0.081	0.028	0.116	0.087	0.031	0.123	0.105	0.096
kPCA-SVR	0.077	0.069	0.136	0.045	0.039	0.144	0.079	0.026	0.130	0.088	0.032	0.111	0.099	0.083
pPCA-SVR	0.089	0.066	0.128	0.047	0.040	0.136	0.077	0.027	0.128	0.088	0.031	0.096	0.097	0.081
DM-SVR	0.088	0.067	0.121	0.047	0.040	0.133	0.078	0.026	0.124	0.089	0.031	0.096	0.095	0.080
FA-SVR	0.144	0.062	0.137	0.041	0.039	0.144	0.066	0.026	0.144	0.085	0.030	0.098	0.082	0.084
PCA-NaNI	0.091	0.080	0.137	0.054	0.045	0.149	0.092	0.029	0.144	0.107	0.032	0.104	0.107	0.090
LDA-NaNI	0.263	0.098	0.209	0.071	0.046	0.216	0.117	0.031	0.149	0.124	0.033	0.158	0.128	0.126
kPCA-NaNI	0.083	0.082	0.159	0.056	0.042	0.154	0.095	0.029	0.160	0.116	0.033	0.125	0.108	0.096
pPCA-NaNI	0.092	0.078	0.139	0.050	0.041	0.148	0.090	0.028	0.139	0.106	0.034	0.105	0.106	0.089
DM-NaNI	0.094	0.080	0.139	0.052	0.043	0.146	0.091	0.026	0.143	0.107	0.030	0.107	0.103	0.089
FA-NaNI	0.152	0.070	0.157	0.046	0.041	0.164	0.075	0.028	0.159	0.098	0.033	0.102	0.087	0.093
PCA-NeNI	0.099	0.093	0.163	0.060	0.047	0.177	0.106	0.030	0.162	0.123	0.035	0.121	0.121	0.103
LDA-NeNI	0.252	0.100	0.194	0.060	0.042	0.217	0.109	0.031	0.151	0.120	0.037	0.158	0.115	0.122
kPCA-NeNI	0.092	0.096	0.187	0.060	0.042	0.175	0.110	0.025	0.180	0.128	0.029	0.135	0.124	0.106
pPCA-NeNI	0.103	0.088	0.162	0.059	0.042	0.170	0.107	0.027	0.160	0.123	0.034	0.120	0.117	0.101
DM-NeNI	0.110	0.090	0.161	0.059	0.046	0.175	0.101	0.025	0.159	0.124	0.034	0.122	0.116	0.102
FA-NeNI	0.176	0.082	0.171	0.054	0.041	0.193	0.087	0.028	0.205	0.114	0.034	0.138	0.096	0.109
PCA-LI	0.092	0.078	0.141	0.055	0.042	0.149	0.095	0.026	0.143	0.114	0.033	0.108	0.108	0.091
LDA-LI	0.254	0.097	0.195	0.062	0.043	0.209	0.107	0.032	0.153	0.115	0.037	0.155	0.113	0.121
kPCA-LI	0.083	0.082	0.159	0.058	0.039	0.159	0.102	0.028	0.160	0.114	0.030	0.127	0.115	0.096
pPCA-LI	0.091	0.080	0.138	0.053	0.047	0.148	0.095	0.029	0.146	0.112	0.034	0.108	0.107	0.091
DM-LI	0.098	0.076	0.142	0.051	0.045	0.149	0.089	0.030	0.146	0.112	0.033	0.108	0.105	0.091
FA-LI	0.160	0.070	0.153	0.046	0.041	0.172	0.078	0.028	0.176	0.102	0.032	0.119	0.087	0.097
sAE(2-Layer)	0.073	0.046	0.126	0.039	0.027	0.149	0.067	0.014	0.123	0.091	0.017	0.099	0.096	0.074

Table 5. Mean reconstruction error per parameter using combinations of dimensionality reduction and reconstruction techniques for the weighted parameterers, with the lowest reconstruction error highlighted in grey. The final column shows the mean (μ) error across all techniques, while the model with the lowest mean reconstruction error (Stacked Autoencoder) is highlighted in green.

P:	0	1	2	3	4	5	6	7	8	9	10	11	12	μ
PCA-LR	0.052	0.059	0.062	0.040	0.023	0.114	0.075	0.018	0.107	0.088	0.020	0.034	0.106	0.061
LDA-LR	0.149	0.068	0.116	0.047	0.022	0.118	0.083	0.017	0.101	0.088	0.020	0.028	0.105	0.074
kPCA-LR	0.039	0.066	0.056	0.043	0.021	0.113	0.084	0.016	0.112	0.089	0.021	0.035	0.105	0.062
pPCA-LR	0.054	0.066	0.062	0.042	0.022	0.111	0.074	0.017	0.108	0.090	0.022	0.036	0.110	0.063
DM-LR	0.058	0.068	0.066	0.041	0.023	0.111	0.074	0.016	0.110	0.091	0.020	0.036	0.107	0.063
FA-LR	0.149	0.062	0.141	0.035	0.021	0.111	0.063	0.015	0.066	0.075	0.022	0.024	0.091	0.067
PCA-SVR	0.046	0.059	0.059	0.041	0.021	0.111	0.071	0.015	0.103	0.087	0.021	0.035	0.099	0.059
LDA-SVR	0.155	0.070	0.120	0.047	0.023	0.121	0.081	0.016	0.109	0.094	0.020	0.027	0.104	0.076
kPCA-SVR	0.036	0.068	0.052	0.044	0.023	0.111	0.080	0.016	0.106	0.090	0.022	0.035	0.108	0.061
pPCA-SVR	0.047	0.061	0.058	0.041	0.023	0.113	0.074	0.016	0.106	0.094	0.021	0.035	0.101	0.061
DM-SVR	0.050	0.063	0.060	0.042	0.024	0.110	0.074	0.016	0.103	0.089	0.020	0.035	0.100	0.060
FA-SVR	0.141	0.050	0.136	0.036	0.023	0.108	0.058	0.017	0.064	0.075	0.019	0.024	0.092	0.065
PCA-NaNI	0.048	0.066	0.064	0.047	0.026	0.127	0.081	0.019	0.116	0.096	0.024	0.038	0.111	0.066
LDA-NaNI	0.195	0.092	0.152	0.062	0.025	0.160	0.106	0.020	0.135	0.123	0.026	0.033	0.123	0.096
kPCA-NaNI	0.038	0.075	0.061	0.051	0.026	0.137	0.098	0.020	0.120	0.102	0.024	0.039	0.110	0.069
pPCA-NaNI	0.046	0.065	0.064	0.045	0.027	0.128	0.080	0.022	0.117	0.094	0.021	0.036	0.110	0.066
DM-NaNI	0.054	0.070	0.069	0.046	0.028	0.128	0.084	0.019	0.118	0.100	0.024	0.038	0.109	0.068
FA-NaNI	0.164	0.055	0.163	0.040	0.023	0.124	0.069	0.019	0.077	0.090	0.025	0.029	0.104	0.076
PCA-NeNI	0.057	0.077	0.080	0.057	0.029	0.157	0.100	0.022	0.140	0.119	0.022	0.043	0.126	0.079
LDA-NeNI	0.195	0.096	0.157	0.063	0.027	0.157	0.105	0.023	0.132	0.122	0.027	0.032	0.123	0.097
kPCA-NeNI	0.042	0.081	0.072	0.058	0.030	0.154	0.108	0.024	0.145	0.112	0.025	0.045	0.125	0.079
pPCA-NeNI	0.054	0.072	0.076	0.055	0.027	0.155	0.097	0.022	0.137	0.110	0.022	0.042	0.130	0.077
DM-NeNI	0.059	0.075	0.084	0.053	0.030	0.158	0.095	0.022	0.143	0.114	0.025	0.045	0.129	0.079
FA-NeNI	0.185	0.064	0.190	0.047	0.029	0.144	0.085	0.020	0.091	0.109	0.025	0.033	0.117	0.088
PCA-LI	0.052	0.070	0.069	0.050	0.027	0.136	0.087	0.021	0.127	0.102	0.026	0.038	0.119	0.071
LDA-LI	0.192	0.103	0.154	0.062	0.027	0.161	0.110	0.018	0.140	0.135	0.025	0.035	0.124	0.099
kPCA-LI	0.037	0.069	0.064	0.049	0.027	0.138	0.094	0.020	0.122	0.106	0.024	0.040	0.113	0.069
pPCA-LI	0.052	0.071	0.069	0.049	0.026	0.137	0.084	0.020	0.125	0.102	0.024	0.039	0.116	0.070
DM-LI	0.054	0.070	0.070	0.046	0.029	0.132	0.085	0.020	0.121	0.099	0.024	0.037	0.113	0.069
FA-LI	0.170	0.056	0.162	0.040	0.026	0.124	0.070	0.021	0.077	0.093	0.025	0.030	0.103	0.077
sAE(3-Layer)	0.065	0.053	0.081	0.040	0.021	0.106	0.075	0.015	0.077	0.081	0.017	0.028	0.096	0.058

Finally, the parameter weighting stage improves the *trustworthiness* of the low-dimensional mapping when using PCA, pPCA, kPCA, DM, and sAE, whilst FA and LDA exhibited significantly lower scores, as presented in Table 6. On the other hand, the *continuity* of the systems had very little change, with pPCA, kPCA, DM, FA, and sAE showing very minor reductions, LDA showing significant reduction, and PCA showing an improvement. In this case, sAE with parameter weighting still outperforms the other techniques in terms of *trustworthiness* for a lower number of neighbours, as in Figure 5c, and the performance in terms of continuity sees the sAE performing better than FA (Figure 5d).

Table 6. *Trustworthiness* and *continuity* scores (including weighting) for the different dimensionality reduction techniques (higher values are better), and classification accuracy of 1-nn classification

Technique	Trustworthiness	Continuity	1-NN Classification
Original	-	-	84.9%
PCA	0.8463	0.9562	67.85%
pPCA	0.8454	0.9552	67.39%
kPCA	0.8263	0.9566	69.40%
FA	0.7761	0.9359	59.52%
DM	0.8477	0.9561	66.03%
LDA	0.6702	0.8340	73.92%
sAE(3-Layer)	0.8440	0.9431	73.51%

5.4. User Evaluation

We evaluate the performance of the selected mode (sAE) using subjective tests in which we present the user with various samples and ask them to equalise it using the low-dimensional space (shown in Figure 6). We then measure the class separability using the JMD metric presented in Section 3.2. In Table 7 we present the degree of separation between user inputs using high-dimensional and low-dimensional responses from the subjective data. From this we can deduce that the overlap between *warm* and *bright* descriptors has decreased, with a value of 0.8527. This is higher than the high-dimensional dataset instances (0.5581). Furthermore, we see an increase in separation between the high-dimensional classes and the opposing low-dimensional classes. For instance, the high-dimensional *warm* examples and the low-dimensional *bright* examples achieve a separation of 0.7719, again higher than the original separation between the high-dimensional classes. Similarly, a strong positive correlation between high-dimensional and low-dimensional equalisation is exhibited by examples in the same class, a desired effect that displays the ability of the users to choose the corresponding regions for the two descriptors.

Table 7. Jeffries-Matusita Distance (JMD) scores showing separation for data gathered from 13-dimensional parameters and a two-dimensional interface using *warm*(W) and *bright*(B) examples. Higher scores are desirable for the first four measurements, while lower scores are better for the last two columns.

Separability	W(13-d)/B(13-d)	W(2-d)/B(2-d)	W(13-d)/B(2-d)	B(13-d)/W(2-d)	W(13-d)/W(2-d)	B(13-d)/B(2-d)
JMD	0.5581	0.8527	0.7719	0.6988	0.0846	0.1439

Table 8. Pearson correlation between the reconstructed equaliser curves.

Metric	B(13-d)/B(2-d)	W(13-d)/W(2-d)	W(13-d)/B(13-d)	W(2-d)/B(2-d)
Pearson correlation	0.9346	0.9247	-0.7594	-0.9121

This is reinforced by the low Euclidean distances between class centroids (shown in Figure 6) and strong positive coherence (spectral correlation) between the equaliser curves achieved using the 13-

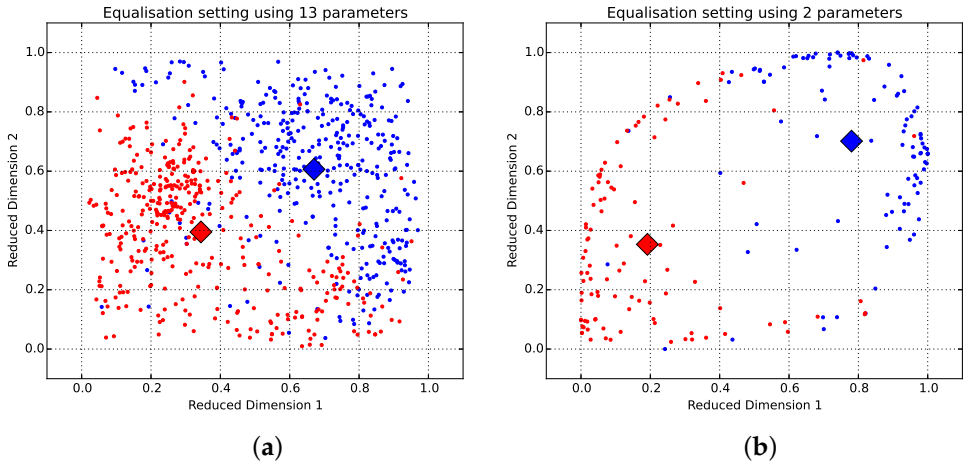


Figure 6. Equalisation settings shown in reduced dimensionality space where the figure (**a**) shows the results of users recording warm and bright samples using 13 parameters; (**b**) the results of users producing the same descriptors using a sAE-based two-dimensional equaliser. Here, diamonds represent the class centroids.

and 2-dimensional interfaces (shown in Figure 7a,b). These results are provided through the Pearson correlation measures in Table 8, revealing a positive correlation between the high-dimensional and low-dimensional datasets for the same descriptor: 0.9346 for warm and 0.9247 for bright, and a negative correlation between opposite high-dimensional and low dimensional descriptors: -0.7594 and -0.9121, respectively.

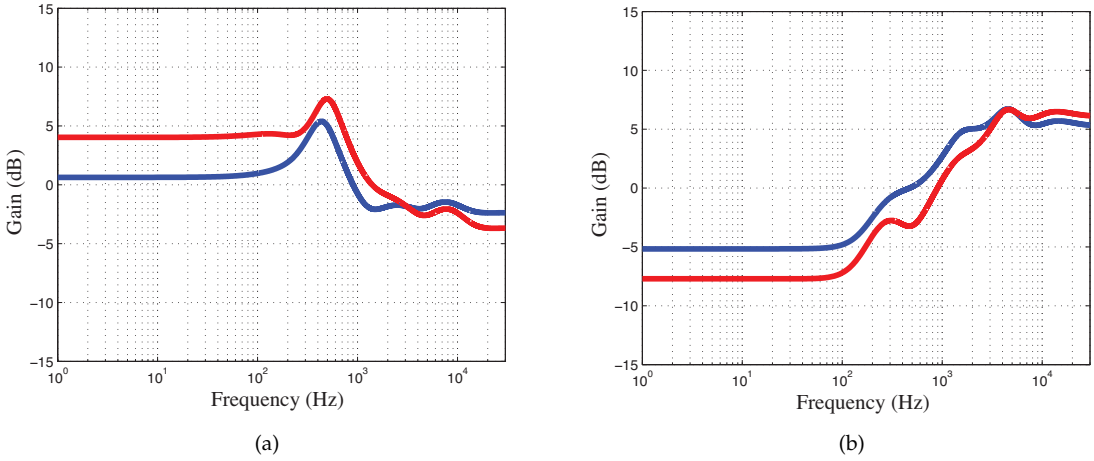


Figure 7. The reconstructed equaliser curves for the centroid of the *warm* and *bright* descriptors for both the high-dimensional (red) and low-dimensional (blue) datasets. (**a**) Reconstructed *warm* equaliser curve; (**b**) Reconstructed *bright* equaliser curve.

6. Discussion

For reconstruction accuracy we find that the sAE is able to outperform all pairwise combinations of dimensionality reduction and reconstruction techniques, whether the system includes parameter weighting or not (Table 4 and Table 5). Furthermore, the sAE is able to achieve the second highest *trustworthiness* score (see Table 2, Figure 5a,c) in low-dimensional space, and performs to a high standard in the preservation of high-dimensional clusters (*continuity*), as in Table 2 and Figure 5b,d.

Using a sAE however, the class-separability in low-dimensional space is reduced when parameter weighting is applied. Furthermore, the system is able to reconstruct the most parameters of the equaliser accurately (six for the unweighted parameters and five for the weighted parameters), while FA with SVR is the only combination able to accurately reconstruct five parameters for the weighted reconstruction. It achieves lower results for overall reconstruction accuracy (0.065), trustworthiness (0.7761), and classification (59.52%), and marginally lower for continuity (0.9359).

Whilst the parameter reconstruction of the autoencoder is sufficiently accurate for our application, it is bound by the intrinsic dimensionality of the data, defined as the minimum number of variables required to accurately represent the variance in lower dimensional space. For the *bright*/*warm* parameter-space data used in this experiment, we can show that the intrinsic dimensionality requires three variables when computed using Maximum Likelihood Estimation [47]. As our application requires a two-dimensional interface, this means the reconstruction accuracy is inherently limited.

Additionally, the user tests revealed that the two-dimensional slider using a sAE is able to accurately reconstruct the equaliser curve, retaining the characteristics associated with *warm* (boost on low-mid and cut on high-end) and *bright* (cut on low-end and boost on high-end), as displayed in Figure 7a,b. Participants of the experiment also commented that the underlying two-dimensional map is easy to quickly learn and provides an intuitive tool for controlling an audio equaliser. Taking into account that the final audio effect should be incorporated alongside the equaliser, with the high-dimensional parameters also available to the users, and with indications as to where the semantic regions are placed, it can be expected that the resulting effect will feature a quick way of achieving the different descriptors (using the two-dimensional slider) and a further fine-tuning stage (via changing the high-dimensional equaliser parameters) if that is necessary.

Providing the model training is applied offline, mapping techniques such as PCA, LDA, DM, pPCA, kPCA, and FA are all capable of running in real-time given the lower degree of computational complexity, as do reconstruction methods such as the interpolation techniques (LI, NaNI, NeNI) and the sAE. Similarly, while the sAE requires iterative training, which will have variable training times based on the number of iterations, the learning rate and the number of neurons and hidden layers, it still offers a fast implementation as the user-input process is relatively lightweight.

7. Conclusions

We have presented a model for the modulation of equalisation parameters using a two-dimensional control interface. The model utilises a sAE to modify the dimensionality of the input data and a weighting process that adapts the parameters to the LTAS of the input audio signal. We train the model with semantics data in order to get the appropriate decoder weights and bias units, which can then be applied to any new input data. This data is given by a user as the position of the cursor changes in an (x,y) Cartesian space. This new information will compute high-dimensional values, which will be rescaled and unweighted, and consequently passed to the equaliser parameters. We show that the sAE model achieves better reconstruction accuracy than other regression and interpolation techniques, achieving an error as low as 0.058. Similarly, the *trustworthiness* and *continuity* of the system perform similarly to (and in some cases outperform) the rest of the dimensionality reduction techniques. Through subjective testing, we can show that the 2D equaliser provides users with an intuitive tool to recreate the high-dimensional equaliser settings extracted from the original dataset. This is demonstrated by comparing the centroids taken from the high and low-dimensional maps and by comparing the equalisation curves when applied to *warm* and *bright* samples.

Acknowledgments: The work of the first author is supported by The Alexander S. Onassis Public Benefit Foundation.

Author Contributions: The work was done in close collaboration. Spyridon Stasis conducted the experiments, derived results and contributed to the manuscript. Ryan Stables defined the mathematical models and drafted sections of the manuscript. Jason Hockman co-developed the models and contributed to the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Valimaki, V.; Reiss, J. All About Audio Equalization: Solutions and Frontiers. *Appl. Sci.*, unpublished work, 2016.
2. Bazil, E. *Sound Equalization Tips and Tricks*; PC Publishing, Norfolk, UK, 2009.
3. Sarkar, M.; Vercoe, B.; Yang, Y. Words that describe timbre: A study of auditory perception through language. In Proceedings of the 2007 Language and Music as Cognitive Systems Conference, Cambridge, UK, 11–13 May 2007; pp. 11–13.
4. Beauchamp, J.W. Synthesis by spectral amplitude and "Brightness" matching of analyzed musical instrument tones. *J. Audio Eng. Soc.* **1982**, *30*, 396–406.
5. Schubert, E.; Wolfe, J. Does timbral brightness scale with frequency and spectral centroid? *Acta Acust. United Acust.* **2006**, *92*, 820–825.
6. Marozeau, J.; de Cheveigné, A. The effect of fundamental frequency on the brightness dimension of timbre. *J. Acoust. Soc. Am.* **2007**, *121*, 383–387.
7. Grey, J.M. Multidimensional perceptual scaling of musical timbres. *J. Acoust. Soc. Am.* **1977**, *61*, 1270–1277.
8. Zacharakis, A.; Pastiadis, K.; Reiss, J.D.; Papadelis, G. Analysis of musical timbre semantics through metric and non-metric data reduction techniques. In Proceedings of the 12th International Conference on Music Perception and Cognition, Thessaloniki, Greece, 23–28 July 2012; pp. 1177–1182.
9. Brookes, T.; Williams, D. Perceptually-motivated audio morphing: Brightness. In Proceedings of the 122nd Convention of the Audio Engineering Society, Vienna, Austria, 5–8 May 2007.
10. Zacharakis, A.; Reiss, J. An additive synthesis technique for independent modification of the auditory perceptions of brightness and warmth. In Proceedings of the 130th Convention of the Audio Engineering Society, London, UK, 13–16 May 2011.
11. Hafezi, S.; Reiss, J.D. Autonomous multitrack equalization based on masking reduction. *J. Audio Eng. Soc.* **2015**, *63*, 312–323.
12. Perez-Gonzalez, E.; Reiss, J. Automatic equalization of multichannel audio using cross-adaptive methods. In Proceedings of the 127th Convention of the Audio Engineering Society, New York, USA, 9–12 October 2009.
13. Cartwright, M.; Pardo, B. Social-EQ: Crowdsourcing an equalization descriptor map. In Proceedings of the 14th ISMIR Conference, Curitiba, Brazil, 4–8 November 2013; pp. 395–400.
14. Mecklenburg, S.; Loviscach, J. SubjEQt: Controlling an equalizer through subjective terms. In Proceedings of the CHI-06, Montreal, QC, Canada, 22–27 April 2006; pp. 1109–1114.
15. Sabin, A.T.; Pardo, B. 2DEQ: An intuitive audio equalizer. In Proceedings of the 7th ACM Conference on Creativity and Cognition, Berkeley, CA, USA, 27–30 October 2009; pp. 435–436.
16. Bristow-Johnson, R. Cookbook formulae for audio EQ biquad filter coefficients. Available online: http://www.musicdsp.org/files/Audio-EQ-Cookbook.txt (accessed on 25 February 2016).
17. Verfaille, V.; Arfib, D. A-DAFx: Adaptive digital audio effects. In Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-01), Limerick, Ireland, 6–8 December 2001.
18. Verfaille, V.; Zölzer, U.; Arfib, D. Adaptive digital audio effects (A-DAFx): A new class of sound transformations. *IEEE Trans. Audio Speech Lang. Process.* **2006**, *14*, 1817–1831.
19. Zölzer, U.; Amatriain, X.; Arfib, D. *DAFX: Digital Audio Effects*; Wiley Online Library: New York, NY, USA, 2011.
20. Stables, R.; Enderby, S.; de Man, B.; Fazekas, G.; Reiss, J.D. SAFE: A system for the extraction and retrieval of semantic audio descriptors. In Proceedings of the 15th ISMIR Conference, Taipei, Taiwan, 27–31 October 2014.
21. Semantic Audio: The SAFE Project. Available online: http://www.semanticaudio.co.uk/ (accessed on 20 January 2016).

22. Brookes, T.; Williams, D. Perceptually-motivated audio morphing: Warmth. In Proceedings of the 128th Convention of the Audio Engineering Society, London, UK, 22–25 May 2010.
23. Venna, J.; Kaski, S. Local multidimensional scaling with controlled tradeoff between trustworthiness and continuity. In Proceedings of 5th Workshop on Self-Organizing Maps, Paris, France, 5–8 September 2005; pp. 695–702.
24. Van der Maaten, L.J.P.; Postma, E.O.; van den Herik, H.J. Dimensionality reduction: A comparative review. *J. Mach. Learn. Res.* **2009**, *10*, 66–71.
25. Sanguinetti, G. Dimensionality reduction of clustered data sets. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 535–540.
26. Stasis, S.; Stables, R.; Hockman, J. A model for adaptive reduced-dimensionality equalisation. In Proceedings of the 18th International Conference on Digital Audio Effects, Trondheim, Norway, 30 November–3 December 2015.
27. Chaudhuri, K.; McGregor, A. Finding metric structure in information theoretic clustering. *COLT Citeseer* **2008**, *8*. Available online: https://people.cs.umass.edu/ mcgregor/papers/08-colt.pdf (accessed on 20 April 2016).
28. Johnson, D.; Sinanovic, S. Symmetrizing the kullback-leibler distance, Computer and Information Technology Institute, Department of Electrical and Computer Engineering, Rice University, Houston, Texas, USA, 2001. Available online: http://www.ece.rice.edu/~dhj/resistor.pdf (accessed on 10 March 2016).
29. Bruzzone, L.; Roli, F.; Serpico, S.B. An extension of the Jeffreys-Matusita distance to multiclass cases for feature selection. *IEEE Trans. Geosci. Remote Sens.* **1995**, *33*, 1318–1321.
30. Senior,M. Mixing secrets for the small studio additional resources. Available online: http://www.cambridge-mt.com/ms-mtk.htm (accessed on 20 January 2016).
31. Van der Maaten, L.J.P. An introduction to dimensionality reduction using Matlab. *Report* **2007**, *1201*. Available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.107.1327&rep=rep1&type=pdf (accessed on 20 April 2016).
32. Hotelling, H. Analysis of a complex of statistical variables into principal components. *J. Educ. Psychol.* **1933**, *24*, 417–441.
33. Schölkopf, B.; Smola, A.; Müller, K.R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.* **1998**, *10*, 1299–1319.
34. Bilmes, J.A. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. *Int. Comput. Sci. Inst.* **1998**, *4*. Available online: http://lasa.epfl.ch/teaching/lectures/ML_Phd/Notes/GP-GMM.pdf (accessed on 20 April 2016).
35. Roweis, S. EM algorithms for PCA and SPCA. *Adv. Neural Inf. Process. Syst.* **1998**, 626–632.
36. Khosla, N. Dimensionality Reduction Using Factor Analysis. Ph.D. Thesis, Griffith University, Brisbane, Queensland, Australia, December 2004.
37. Nadler, B.; Lafon, S.; Coifman, R.R.; Kevrekidis, I.G. Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. *Appl. Comput. Harmonic Anal.* **2006**, *21*, 113–127.
38. Fisher, R.A. The use of multiple measurements in taxonomic problems. *Ann. Eugen.* **1936**, *7*, 179–188.
39. Bobach, T.; Umlauf, G. Natural neighbor interpolation and order of continuity. University of Kaiserslautern, Computer Science Department/IRTG, Kaiserslautern, Germany, 2006. Available online: http://www-umlauf.informatik.uni-kl.de/~bobach/work/publications/dagstuhl06.pdf (accesed on 20 January 2016).
40. Drucker, H.; Burges, C.J.C.; Kaufman, L.; Smola, A.; Vapnik, V. Support vector regression machines. *Adv. Neural Inf. Process. Syst.* **1997**, *9*, 155–161.
41. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507.
42. Bengio, Y. Learning deep architectures for AI. *Found. Trends Mach. Learn.* **2009**, *2*, 1–127.
43. Bergstra, J.; Breuleux, O.; Bastien, F.; Lamblin, P.; Pascanu, R.; Desjardins, G.; Turian, J.; Warde-Farley, D.; Bengio, Y. Theano: A CPU and GPU math compiler in Python. In Proceedings of the 9th Python in Science Conference (SciPy), Austin, Texas, USA, 28 June–3 July 2010.
44. Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; Manzagol, P.A. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.* **2010**, *11*, 3371–3408.

45. Tieleman, T.; Hinton, G. Lecture 6e - rmsprop: Divide the gradient by a running average of its recent magnitude, 2012. Available online: http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf (accessed on 20 January 2016).
46. Dauphin, Y.; de Vries, H.; Bengio, Y. Equilibrated adaptive learning rates for non-convex optimization. In Proceedings of Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015, pp. 1504–1512.
47. Levina, E.; Bickel, P.J. Maximum likelihood estimation of intrinsic dimension. In Proceeding of Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 13–18 December 2004.

© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons by Attribution (CC-BY) license (http://creativecommons.org/licenses/by/4.0/).