# VOICE FEATURES FOR CONTROL: A VOCALIST DEPENDENT METHOD FOR NOISE MEASUREMENT AND INDEPENDENT SIGNALS COMPUTATION

*Stefano Fasciani*

Graduate School for Integrative Sciences & Engineering
Arts and Creativity Laboratory, Interactive and Digital Media Institute
National University of Singapore
stefano17@nus.edu.sg

**ABSTRACT**

Information about the human spoken and singing voice is conveyed through the articulations of the individual's vocal folds and vocal tract. The signal receiver, either human or machine, works at different levels of abstraction to extract and interpret only the relevant context specific information needed. Traditionally in the field of human machine interaction, the human voice is used to drive and control events that are discrete in terms of time and value. We propose to use the voice as a source of real-valued and time-continuous control signals that can be employed to interact with any multidimensional human-controllable device in real-time. The isolation of noise sources and the independence of the control dimensions play a central role. Their dependency on individual voice represents an additional challenge. In this paper we introduce a method to compute case specific independent signals from the vocal sound, together with an individual study of features computation and selection for noise rejection.

## 1. INTRODUCTION

The human voice is an extremely flexible sound generation mechanism and we use it primarily to transfer different categories of information through acoustic communications. The human brain interprets and understands it transparently, while for machines the task is challenging. The processing of the vocal signal, in most human machine application domains, begins with a stage of feature computations to obtain a compact representation of the audio signal, and is followed by the application of one or more statistical models to decode information. Other than well established and commonly used applications such as speech recognition, speaker identification, voice detection, music information retrieval (querying by voice) and voice transformation, there are more recent applications domains in which the interaction is established at sub-verbal level.

The resulting interaction, when working directly with low level features of the voice, is more direct and immediate [1]. The "vocal joystick", presented in [2] and studied in [3], computes energy, pitch and vowel quality (vowel recognition) to provide a 2-D pointer navigation system, underlying the importance of independence in vocal features selection. Extensions of this work are "VoiceDraw" [4] and the "VoiceBot" [5], where a similar technique is applied to screen drawing and to the manipulation of a robotic arm with 5 degree of freedom. Motor-impaired subjects see these voice controlled interfaces as an accessible system, while others see these as a hands free extension to traditional controllers. However the works mentioned above, even if working at sub-verbal level, still present limitations due to the presence of classifiers driving discrete events.

In [6] and [7] a richer set of low level vocal features, including the MFCC, are used to build a more complex voice controlled interface for a wah-wah pedal and for an audio mosaicing synthesizer respectively. The Gesture Follower[1] and the Wekinator [8] are two machine learning based system for mapping generic human gesture to real-valued continuous parameters of any controllable device. The latter two are not designed to control any particular class of devices, but they can implement any nature of sub-verbal interface, computing low-level vocal features as a source of gestural input, but noise issues and independence of the control dimension remain largely unaddressed.

In this paper we present a generic method to analyze the low-level features of the voice with the aim of improving the robustness and the control capabilities of any vocal control system that does not make use of classification techniques. The method can be applied indiscriminately on spoken voice, singing voice and pure sub-verbal sounds. We developed this generic technique to improve the control capabilities of the sub-verbal vocal interface for digital musical instruments that we proposed in [9].

The characteristics of the vocal folds and the vocal tract present high variability among different speakers, as do speaking or singing style and different individual native languages. The proposed method minimizes the noise and maximizes the independence of the computed control signals over specific performances of individual vocalists, rather than attempt to provide vocalist-independent study results, which sacrifice optimization for generalization. In Section 2 we present the method for noise features rejection and independence measurements starting from the computation of a large feature set. Experimental result, on different vocalists in different environmental conditions, are described and in Section 3. Conclusions and usability issues are discussed in Section 4.

## 2. NOISE AND INDEPENDENCE MEASUREMENT

Within this work we adapt some terminology from the non-verbal communication literature, defining: "vocal posture" as the action of uttering sound with invariant characteristics over time, and "vocal gesture" as the action of uttering sound with characteristics varying over time. We ground the study of noise and independence for the low level features computed from the vocal audio signal on two generic requirements for the sub-verbal interface:

- The control signals computed over a vocal posture must have constant values (static behaviour).
- The control signals computed over a vocal gesture must not be redundant (dynamic behaviour).

[1] http://ftm.ircam.fr/index.php/Gesture_Follower

The absence of voice can be considered as a special case of vocal posture. The inhibition of the control system when no voice is present at the input can be considered as a further requirement. There are techniques described in the literature which can be exploited for this purpose that are capable of detecting voice even in challenging conditions such as the concurrent presence of voice and music [10] [11].

The main issue related to vocal postures is the presence of noise. Even though a vocalist has the perception of uttering invariant sound over time, such as sustained vowels, some low-level features may still have a significant variance. Features affected by this noise are different across individuals. Therefore we define noisy features as those with a statistical dispersion above a certain threshold when computed over a set of vocal postures from a specific vocalist. An arbitrary number of control signals computed from a vocal gesture (selection or transformations of the low-level features) should present statistical independence, or at least low correlation. If not, the multiple control dimension, presumably mapped over different device parameters, would vary similarly to one another, providing a trivial control system. The vocal folds and the vocal tract can be approximated with a source-filter system. These two systems are independent and within each system there are independent sub-components as well: energy and pitch in the vocal folds, and the first two formants frequencies in the vocal tract representing the vowel space, just to mention a few. But given a specific vocal gesture, this independence assumption may not be valid anymore. In our approach, we do not use any prior knowledge about the independence of the vocal features, but we perform a posterior study, based on a vocalist-specific gesture (speech, singing or sub-verbal sounds) using a method to find independence from the non-noisy features retained in the system. In [12] Stowell and Plumber present a study on the degradation and independence of voice timbre features subjected to acoustic degradations, providing a sort of feature ranking for general purpose usage. They provide vocalist-independent results, while defining three different vocal categories: singing, speaking and beatboxing. As mentioned above, in our approach the study is based on a specific individual voice and a specific performance, without introducing any categorization. We extend the study on a larger feature set and we explore several computing parameters combination. Moreover the key concept of feature robustness is substantially different from previous work to address a specific feature computation purpose, which is the generation of robust, independent, time and value continuous control signals.

### 2.1. Parametric Low Level Features Computation

Since we assume no prior knowledge about the vocalist's voice characteristics and the kind of vocal gesture used for control purposes, we initially compute a large set of features, including all those commonly used in speech processing applications. This feature set may present high redundancy, but we leave the feature selection to a following stage. The features computed are:

- Energy;
- Pitch;
- Linear Predictive Coding coefficients (LPC);
- Mel Frequency Cepstrum Coefficients (MFCC);
- Perceptual Linear Predictive coefficients (PLP) [13];
- RelAtive SpecTrAl Perceptual Linear Predictive coefficients (RASTA-PLP) [14];
- Delta coefficients;
- Delta-delta (acceleration) coefficients.

The computation and the post processing of the low level features are implemented in MATLAB. Pitch, LPC and MFCC are computed using the Voicebox[2] package; PLP and RASTA-PLP are computed using the Rastamat[3] package. Within the feature computation process there are several parameters to choose. These affect aspects of the eventual real-time voice controlled interface, such as the latency, the time resolution and the computational cost. At the same time they may affect the noise and independence of the control signals computed from the voice. Instead of choosing fixed parameters we perform a systematic study, testing different combinations of features computation parameters and picking the one resulting in the better performances. It has been previously shown [15] that even in a different application domain such as speaker independent speech recognition, the optimal performances are obtained for different computation parameters once the nature of the feature vector is fixed. In this study, two quality indicators measure the performances: one is computed within the noise detection and rejection phase, the other one after the independence analysis.

The parameters we expose to variation within the features computation are:

- Window size;
- Window overlap;
- Pre-emphasis;
- Order of the various features vectors.

For the window size, we explore the range from 128 to 2046, considering only the values power of two. For the window overlap the tested values are 25%, 50% and 75%. Three are the values for the pre-emphasis: high (0.97=H), mid (0.485=M) and zero (0.0=Z). With order we intend the number of LPC, MFCC, PLP and RASTA-PLP coefficients computed, corresponding to the number of spectral sub-bands for the cepstral coefficients. We vary the order in the range 8 to 16 with a step of 2. We discard the first LPC coefficient because it is constant and the first MFCC and PLP coefficients because they are redundant with the energy. Therefore the number of computed features depends by the order and is equal to:

$$\dim(\mathbf{f}) = (3 \cdot ((order \cdot 4) + 3)) \quad (1)$$

where **f** represents the feature vector computed for each window of the vocal audio signal. The feature computation described above leads to 225 different combinations to optimize over. Wider parameter ranges with a finer step help to find a solution closer to the absolute optimum. Since the optimal parameters depend on the vocalist and the specific gesture, in this paper we aim to present a methodology rather than derive parameters that work across general cases.

The audio sampling rate is a flexible parameter in the system, however for the experiment described in this paper it has been fixed to 16KHz. Such a low audio sampling rate, common in speech processing application, is a trade-off between low computational cost and loss of information at higher frequencies. In a real sub-verbal interface implementation, a low computational complexity of the processing chain is desirable, because the control signals are generated from the vocal signal in real-time. Even if most of the energy is concentrated below the 8KHz Nyquist frequency, singing and speech may have frequency components up to 20KHz. Historical and physical reasons for neglecting the band above 8KHz in speech processing applications are dis-

[2] http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html
[3] http://labrosa.ee.columbia.edu/matlab/rastamat/

cussed in [16], where the author investigates its audibility and perceptual significance.

### 2.2. Robust Features Selection Based On Noise Measurement

We mark a feature as noisy, and thus we discard it, if its statistical dispersion, computed over perceptually vocal posture, is above a certain threshold. As a measurement of statistical dispersion we use the Relative Mean Difference (RMD) (2), because it is scale invariant.

$$RMD = \left( \left| \sum_{i=1}^{n} x_i \right| (n-1) \right)^{-1} \sum_{i=1}^{n} \sum_{j=1}^{n} \left| x_i - x_j \right| \tag{2}$$

In (2) $n$ represents the number of samples, which in our case corresponds to the number of analysis windows over a single vocal posture, while $x$ represents the feature under question. The first summation in (2) comes from the arithmetic mean and taking its absolute value produces only positive RMD values, facilitating the subsequent operations. Given a database of voice recordings, each containing different vocal postures by an individual vocalist, we compute the low level features. For each feature we compute the RMD over a single recording, and we compute the average of the RMDs to measure the statistical dispersion over the whole database. Features with the average RMD over the threshold are marked as noisy and thus discarded.

We ran this study for all the possible combinations of the feature computation parameters. As a Robustness Quality Measure (RQM) of every combination we choose the inverse of the average RMD of all the non-rejected features, normalized by the number of non-noisy features (3),

$$RQM = \left( \mathrm{E}[RMD] / |feat.| \right)^{-1} \tag{3}$$

where *E[RMD]* represents the average of the RMD of the robust features, and $|feat.|$ represents the number of robust features. In (3) we promote cases in which the number of non-rejected features is higher because it may potentially increase the quantity of independent information in the subsequent study. In (3) we compute the inverse to obtain a RQM value growing with the overall quality. This method addresses the individual inability to utter sustained invariant sounds, called vocal postures in this work, even when having the subjective perception of doing so.

### 2.3. Independent Control Signals Computation

The study of the independence is performed over an individual and specific vocal gesture consisting of speech, singing or pure sub-verbal sounds. The independence measurement retrieved from the training examples can be then applied to implement live vocal control, where the vocalist does not necessarily have to stick to the same training gesture in terms of temporal unfolding. We compute the low level features over the training recordings, then we keep only the ones marked as robust from the previous study and we apply the Independent Component Analysis (ICA) method for the independence signals computation. We repeat these measurements for five parameter combinations corresponding to the local maxima of the robustness quality measurement for the five different order ranges.

ICA is a statistical technique assuming a nongaussian distribution of the sources and their statistical independence [17]. In this case we assume that the P robust features $f_i$ are a linear combination of J random variables $s_j$, statistically independent and nongaussian, as in (4). In (5) the linear combination is expressed in matrix notation.

$$f_p = \alpha_{p1} s_1 + \alpha_{p2} s_2 + \ldots + \alpha_{pJ} s_J \quad \text{for } p = 1, \ldots, P \tag{4}$$

$$\mathbf{f} = \mathbf{A}\mathbf{s} \tag{5}$$

$$\mathbf{s} = \mathbf{W}\mathbf{f} \tag{6}$$

In (5) **A** represents the mixing matrix while in (6) **W** is the unmixing matrix. The ICA algorithm iterates until convergence on a matrix **W** that gives the maximally nongaussian sources. The ICA requires a number of observations **f** at least equal to the number of sources **s**. This condition always holds in our study because we consider a maximum number of independent component at least 1 order or magnitude smaller than the number of feature computation windows. We use the FastICA[4] [18] package for the MATLAB computation of the independent component analysis because of its efficiency.

### 2.4. Test and Evaluation Method

A global evaluation method, as well as a metric to compare among the different feature computation parameters combinations, is based on three different testing conditions, from which we compute a triplet of quality parameters $Q_1$, $Q_2$ and $Q_3$. We extract the independent control signals performing the feature computation, robust feature selection and the ICA unmixing on:

- Vocal postures to measure how constant the control signals are ($Q_1$);
- Vocal gestures similar to the one used for the **W** estimation, to measure control signal independency ($Q_2$);;
- Vocal gestures with a different temporal unfolding from the one used for the **W** estimation, to measure control signal independency ($Q_3$);

The vocal recordings used for the testing differ from the ones used for the noise measurement and ICA. In the first test the independent signals are computed from a new set of vocal postures and we measure the RMD for each $s_j$, taking their sum as in (7).

$$Q_1 = \sum_{k=1}^{J} RMD(s_K(t)) \tag{7}$$

In the second and third tests we compute the $s_j$ from a different instance of the same vocal gesture used to estimate **W,** and from others gestures which differ in some aspects. Other than changing the temporal unfolding we tested also other gesture variation such as singing performance presenting different lyrics over the same score, or vice versa. In both tests we measure the independence of the obtained $s_j$. As independence measure we use the distance correlation, defined in (8), (9) and (10), which is a measure of the statistical dependence between two random vectors which do not necessarily share the same dimensionality [19]. A null distance correlation implies independence, while a null correlation implies independence only if the random variables are Gaussians, which is not our case. The distance correlation depends on the distance covariance, equivalent to the Brownian covariance [20].

$$\mathrm{dCor}(X,Y) = \mathrm{dCov}(X,Y) \Big/ \sqrt{\mathrm{dVar}(X) \cdot \mathrm{dVar}(Y)} \tag{8}$$

$$\begin{aligned} \mathrm{dVar}^2(X) := \mathrm{E}[\|X - X'\|^2] + \mathrm{E}^2[\|X - X'\|] + \ldots \\ \ldots - 2\mathrm{E}[\|X - X'\| \cdot \|X - X''\|] \end{aligned} \tag{9}$$

$$\begin{aligned} \mathrm{dCov}^2(X,Y) := \mathrm{cov}\left(\|X - X'\|, \|Y - Y'\|\right) + \ldots \\ \ldots - 2\,\mathrm{cov}\left(\|X - X'\|, \|Y - Y''\|\right) \end{aligned} \tag{10}$$

[4] http://research.ics.tkk.fi/ica/fastica/

In (9), (10) and (11), ||.|| represents the Euclidean norm, while *(X,Y)*, *(X',Y')* and *(X'',Y'')* are independent and identical distributed random variables. We compute the distance correlation $dCor(s_j,S_j)$ between every independent signal $s_j$ and all the remaining, denoted by $S_j$.

$$Q_{2,3} = \sum_{k=1}^{J} \mathrm{dCor}(s_K(t), S_K(t))) \tag{11}$$

From the way we defined the three different quality parameters in (7) and (11), better performance is obtained for small values of the $Q_{1,2,3}$. Hence we define a global quality parameter ISQM (Independent Signals Quality Measurement) in proportion to performances, as the inverse of the sum of $Q_{1,2,3}$, as in (12).

$$ISQM = (Q_1 + Q_2 + Q_3)^{-1} \tag{12}$$

## 3. EXPERIMENTAL RESULTS

In this section we present experimental results based on independent signals computed on different performances of different vocalists. Since the noise and independence study is based on individual vocalists, we do not aim to provide generic results valid across vocalists, but we identify only recurrent trends. We use three different vocalists in our experiments to highlight the capability of this method to adapt itself to the individual voice characteristics, and to highlight differences across optimal settings when changing the subject. The three vocalists, two adult males (Voc.1, Voc.3) and one adult female (Voc.2), differ in their native language, which may influence the speaking or singing style as well. None of them is a professional singer or speaker. We used a MOTU UltraLite recording interface, and selected a different microphone for each vocalist (Crown CM311-A, Shure SM58 and Rode NT55) to increase variation between speakers. The recordings were performed in silent conditions; performance degradations due to noisy recording environment are presented later.

### 3.1. Robustness Analysis

The size of the vocal postures recordings database we use for each vocalist is different (22, 14 and 17 recordings). Each recording has a length of about 3 seconds. The vocal postures are chosen by the vocalist themselves and they may differ across the individual dataset. For the robust feature selection we use the whole database for each vocalist except 2 recordings, randomly chosen, which we use later on for testing purposes. As a threshold value for the feature rejection we choose 0.5, but it can be changed to be compliant with specific application requirements. Figure 1 shows the average RMD and the standard deviation for each feature for the worst and best cases across the 225 computation parameters combination cases, computed on the vocalist 1 database. Each segment in Figure 1 (features in blue, delta in red, and delta-delta in green) contains features in the following order: energy, pitch, LPC, MFCC, PLP, RASTA-PLP. The difference in the x-axis is due to the different order between the worst and best RMD case, which generates feature vectors of different size. As expected, we observe across speakers that the delta and delta-delta differential features are very noisy and therefor not useful for this purpose. In Figure 1 their RMD is not visible when they exceed the value of 10. Energy, pitch, low-order MFCC, PLP and RASTA-PLP are usually more robust than LPC. RASTA-PLP is the most robust features set. In Figure 2 we show the RQM across the 225 cases, computed over 3 vocalist's dataset.

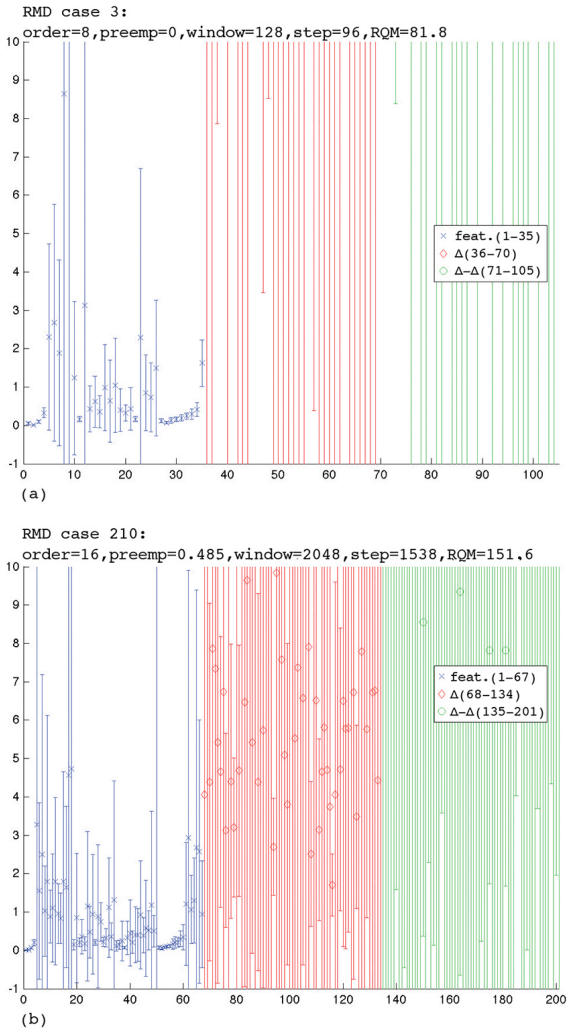


Figure 1: *Worst case (a) and best case (b) features average RMD with standard deviation across the vocalist 1 dataset.*

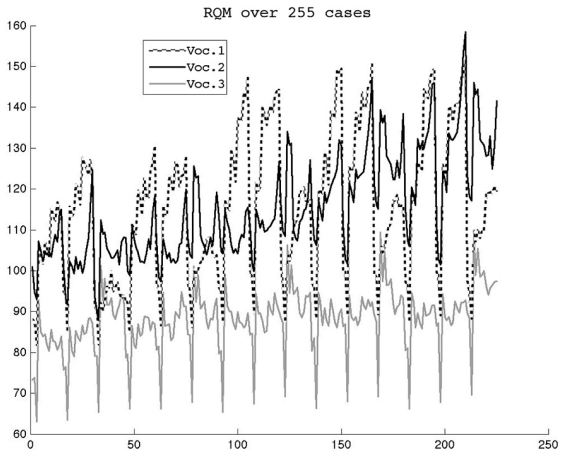


Figure 2: *RQM for three different vocalist dataset over 225 features computation parameters combinations.*

In the computation we iterate over window overlap, window size, pre-emphasis, and features order parameters. The periodicity of the RQM in Figure 2 is due to the specific computation loop nesting. We observe that in general the local maxima for each order range are not shared across vocalists. This, as expected, demonstrates that an individual vocalist approach is necessary to find the optimal features computation parameters. Since RQM promotes cases with a higher number of robust features, we can observe a rising trend due to the increase in the order of the features. In general with higher order the probability of having a higher number of features with the RMD below the threshold is higher, hence we expected and obtained higher RQMs. As we will discuss and present later, the absolute maximum of the RMQ may not coincide with the maximum of $Q_T$. Details of the RQM local maxima for each order are reported in Table 1. In Table 2 the robust features, always for the order's local maxima, are categorized into their class and differential, and as mentioned, it is possible to observe the delta and delta-delta differential features are not robust. Energy and pitch are never rejected, and usually the majority of the robust features belong to the RASTA-PLP group, followed by the MFCC and PLP. However the specific composition of the robust feature vector is different across vocalists, again supporting the individual vocalist approach.

### 3.2. Independent Control Signal Analysis

We compute the ICA over three instances of the same vocal gesture, choosing a number of independent components equal to 4. Each instance has a length of about 10 seconds. As described in 2.4, after computing the unmixing matrix **W**, we run test in three different conditions, computing quality parameter for each case. Since the computational load of the distance correlation is extremely high, we ran these tests using only 20 different parameters configurations for the features computation, coming from the top four RQM over the five different orders. In Table 3 we report the ISQM results, as well as the $Q_{1,2,3}$, obtained for different speech, singing and pure sub-verbal gestures. For the best ISMQ in Table 3, we always obtain a low variance of the independent control signals over vocal postures, while the independence of the $s_j$, measured by the distance correlation, is typically below 0.3. In general we observed that we obtain better performances with a lower feature computation order and with large windows. However the best parameter configuration is vocalist and gesture dependant. In Figure 3 we show the spectrogram versus the independent control signals over a speech gesture for vocalist 1 (a), with the corresponding stable control signals for vocal posture (c), and the best case of singing gesture for vocalist 2 (b).

Table 1: *Robustness Quality Measurement local maxima for each order, with relative feature computation parameters (w=window size, s=window step, p=pre-emphasis) and number of robust features (r.f.), over three vocalists postures dataset. It can be observed the rising trend of the RQM and r.f. with the feature computation order.*

| ***Order*** | ***Voc.1 - RQM*** | ***Voc.2 - RQM*** | ***Voc.3 - RQM*** |
|---|---|---|---|
| **8** | 127.6 (20 r.f.; w=1024; s=25%; p=M) | 124.6 (25 r.f.; w=2048; s=75%; p=M) | 101.2 (19 r.f.; w=256; s=25%; p=H) |
| **10** | 130.3 (26 r.f.; w=2048; s=75%; p=Z) | 125.6 (27 r.f.; w=256; s=25%; p=H) | 101.0 (20 r.f.; w=256; s=25%; p=H) |
| **12** | 147.2 (31 r.f.; w=2048; s=75%; p=Z) | 134.1 (31 r.f.; w=256; s=25; p=H) | 106.3 (21 r.f.; w=256; s=25%; p=H) |
| **14** | 150.1 (32 r.f.; w=2048; s=75%; p=M) | 147.1 (34 r.f.; w=2048; s=75%; p=M) | 109.3 (21 r.f.; w=256; s=25%; p=H) |
| **16** | 151.6 (33 r.f.; w=2048; s=75%; p=M) | 158.4 (36 r.f.; w=2048; s=75%; p=M) | 109.7 (22 r.f.; w=256; s=25%; p=H) |

Table 2: *Percentage of robust featured across features class (capital case columns, where EN.=Energy, PTC.=Pitch, R.PLP=RASTA-PLP) and differential class (lower case columns, where d.=delta, d.d=delta-delta ) for local maxima for each order over three vocalists dataset. RASTA-PLP has higher percentage among the feature classes followed by MFCC and PLP. Energy and Pitch are never rejected. Delta and delta-delta are always marked as noisy in the vocal postures.*

| | ***Order*** | ***EN.*** | ***PTC.*** | ***LPC*** | ***MFCC*** | ***PLP*** | ***R.PLP*** | ***feat.*** | ***d.*** | ***d.d.*** |
|---|---|---|---|---|---|---|---|---|---|---|
| ***Voc.1 %*** | **8** | 5.0 | 5.0 | 10.0 | 15.0 | 25.0 | 45.0 | 100.0 | 0.0 | 0.0 |
| | **10** | 3.8 | 3.8 | 11.5 | 23.0 | 26.9 | 30.7 | 100.0 | 0.0 | 0.0 |
| | **12** | 3.2 | 3.2 | 9.6 | 22.5 | 32.2 | 29.0 | 100.0 | 0.0 | 0.0 |
| | **14** | 3.1 | 3.1 | 6.2 | 25 | 31.2 | 31.2 | 100.0 | 0.0 | 0.0 |
| | **16** | 3.0 | 3.0 | 6.0 | 27.2 | 30.3 | 30.3 | 100.0 | 0.0 | 0.0 |
| ***Voc.2 %*** | **8** | 4.0 | 4.0 | 16.0 | 24.0 | 20.0 | 32.0 | 100.0 | 0.0 | 0.0 |
| | **10** | 3.7 | 3.7 | 7.4 | 29.6 | 22.2 | 33.3 | 100.0 | 0.0 | 0.0 |
| | **12** | 3.2 | 3.2 | 6.4 | 29.0 | 25.8 | 32.2 | 100.0 | 0.0 | 0.0 |
| | **14** | 2.9 | 2.9 | 11.7 | 23.5 | 32.3 | 26.4 | 100.0 | 0.0 | 0.0 |
| | **16** | 2.7 | 2.7 | 11.1 | 25.0 | 30.5 | 27.7 | 100.0 | 0.0 | 0.0 |
| ***Voc.3 %*** | **8** | 5.2 | 5.2 | 5.2 | 21.0 | 21.0 | 42.1 | 100.0 | 0.0 | 0.0 |
| | **10** | 5.0 | 5.0 | 5.0 | 20.0 | 15.0 | 50.0 | 100.0 | 0.0 | 0.0 |
| | **12** | 4.7 | 4.7 | 4.7 | 19.0 | 14.2 | 52.3 | 100.0 | 0.0 | 0.0 |
| | **14** | 4.7 | 4.7 | 4.7 | 19.0 | 14.2 | 52.3 | 100.0 | 0.0 | 0.0 |
| | **16** | 4.5 | 4.5 | 4.5 | 18.1 | 18.1 | 50.0 | 100.0 | 0.0 | 0.0 |

Table 3: *Three independent control signal quality parameters over different vocal gestures (speech, singing and pure sub-verbal), including noisy environment conditions, of different vocalists. The best two ISQM for each gesture are presented with the relative parameters combinations parameters (o=order, w=window size, s=window step, p=pre-emphasis). One can clearly see how the best parameter combination is vocalist and vocal gesture dependant. In general, lower orders and larger windows give the best ISQM. $Q_2$ and $Q_3$ values show consistency in most of the test cases, while $Q_1$ values are usually low.*

| ***Voc.1 Speech*** | | | | | ***Voc.1 Sub-verbal*** | | | | |
|---|---|---|---|---|---|---|---|---|---|
| ***$Q_1$*** | ***$Q_2$*** | ***$Q_3$*** | ***ISQM*** | ***Params.*** | ***$Q_1$*** | ***$Q_2$*** | ***$Q_3$*** | ***ISQM*** | ***Params.*** |
| 0.29 | 0.81 | 0.85 | 0.5 | o=10; w=1024; s=75%; p=M | 0.52 | 0.80 | 0.85 | 0.45 | o=10; w=1024; s=75%; p=M |
| 0.65 | 0.95 | 0.78 | 0.41 | o=14; w=2048; s=75%; p=Z | 0.60 | 0.87 | 0.78 | 0.44 | o=8; w=1024; s=25%; p=M |
| ***Voc.2 Sing*** | | | | | ***Voc.2 Sub-verbal*** | | | | |
| ***$Q_1$*** | ***$Q_2$*** | ***$Q_3$*** | ***ISQM*** | ***Params.*** | ***$Q_1$*** | ***$Q_2$*** | ***$Q_3$*** | ***ISQM*** | ***Params.*** |
| 0.37 | 0.89 | 1.15 | 0.41 | o=8; w=2048; s=50%; p=M | 0.64 | 0.79 | 0.92 | 0.42 | o=12; w=128; s=25%; p=M |
| 0.32 | 1.02 | 1.23 | 0.38 | o=14; w=2048; s=75%; p=M | 0.65 | 1.21 | 1.56 | 0.29 | o=16; w=2048; s=75%; p=M |
| ***Voc.3 Speech*** | | | | | ***Voc.3 Sub-verbal*** | | | | |
| ***$Q_1$*** | ***$Q_2$*** | ***$Q_3$*** | ***ISQM*** | ***Params.*** | ***$Q_1$*** | ***$Q_2$*** | ***$Q_3$*** | ***ISQM*** | ***Params.*** |
| 0.31 | 0.71 | 0.69 | 0.58 | o=12; w=1024; s=75%; p=M | 0.37 | 0.67 | 0.64 | 0.45 | o=12; w=1024; s=75%; p=M |
| 0.47 | 0.98 | 0.88 | 0.42 | o=14; w=1024; s=50%; p=M | 0.53 | 0.89 | 0.94 | 0.41 | o=8; w=1024; s=50%; p=H |
| ***Voc.1 Speech Noisy*** | | | | | ***Voc.2 Sing Noisy*** | | | | |
| ***$Q_1$*** | ***$Q_2$*** | ***$Q_3$*** | ***ISQM*** | ***Params.*** | ***$Q_1$*** | ***$Q_2$*** | ***$Q_3$*** | ***ISQM*** | ***Params.*** |
| 0.18 | 0.94 | 0.88 | 0.49 | o=10; w=512; s=25%; p=H | 0.73 | 0.97 | 1.10 | 0.35 | o=10; w=2048; s=75%; p=Z |
| 0.39 | 0.86 | 0.85 | 0.47 | o=8; w=2048; s=25%; p=H | 0.98 | 0.93 | 1.08 | 0.33 | o=10; w=2048; s=50%; p=M |

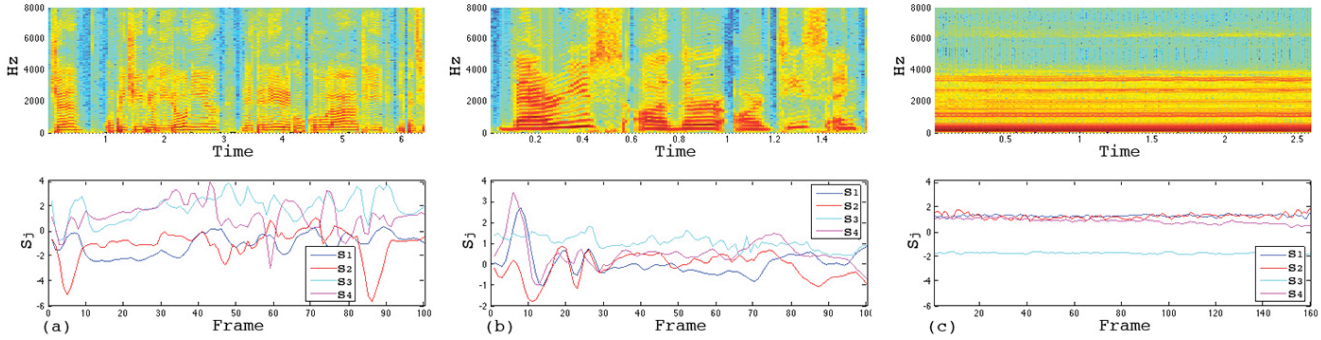


Figure 3: *Three details of spectrogram and independent control signals, obtained by the ICA unmixing matrix, computed on a short interval of two vocal gestures (a) (b) and one vocal posture (c).*

Table 4: *Robustness Quality Measurement local maxima for each order, with relative feature computation parameters (w=window size, s=window step, p=pre-emphasis) and number of robust features (r.f.), over vocalists 1 postures dataset comparing silent and noisy recording conditions. It is possible to observe how the number of robust features and the RQM values are lower, while the feature computation parameters are similar except differences in the step size.*

| ***Order*** | ***Voc.1 - RQM - Silent Env.*** | ***Voc.1 - RQM - Noisy Env.*** |
|---|---|---|
| **8** | 127.6 (20 r.f.; w=1024; s=25%; p=M) | 94.7 (20 r.f.; w=2048; s=50%; p=H) |
| **10** | 130.3 (26 r.f.; w=2048; s=75%; p=Z) | 101.6 (21 r.f.; w=2048; s=50%; p=H) |
| **12** | 147.2 (31 r.f.; w=2048; s=75%; p=Z) | 109.8 (24 r.f.; w=2048; s=75%; p=H) |
| **14** | 150.1 (32 r.f.; w=2048; s=75%; p=M) | 109.3 (28 r.f.; w=2048; s=75%; p=M) |
| **16** | 151.6 (33 r.f.; w=2048; s=75%; p=M) | 110.2 (29 r.f.; w=2048; s=75%; p=M) |

The method, tested with a different number of independent components and with a threshold value different by 0.5, consistently shows the capability to identify the vocalist and gesture dependent configuration to compute the independent control signals with highest independence and lowest noise. In particular, with a less severe threshold, such as 1, the number of robust features increases by about 50%. The threshold determines the trade-off between the value of $Q_1$ and the values of $Q_2$ and $Q_3$. Moreover we observed that in general small orders produces higher $Q_1$, even if the performances on the $Q_2$ and $Q_3$ are slightly better for higher orders.

### 3.3. Noisy Environments Performances

The proposed method presents only a slight performance decrease when the database is recorded in noisy environments. We duplicated the database performing the same vocal gesture and posture recordings in different environmental conditions. Eight loudspeakers simultaneously reproducing music and the microphone signal were surrounding the vocalist. To increase the randomness, the background music was rapidly crossfading across 4 different genre songs every 5 seconds. In Figure 4 we compare the RQM, for vocalist 1, over the 225 different cases, in silent and noisy recording conditions. The RQM decrease in absolute value is evident, while the rising trend as well as the local maxima and minima are similar. In Table 4 we present a comparison of RQM local maxima details, for each order, for the two different recording conditions. Since we did not add artificial noise to the database, but we performed new recordings in a noisy environment, the consistency of the results for the two environments supports the validity of this approach.

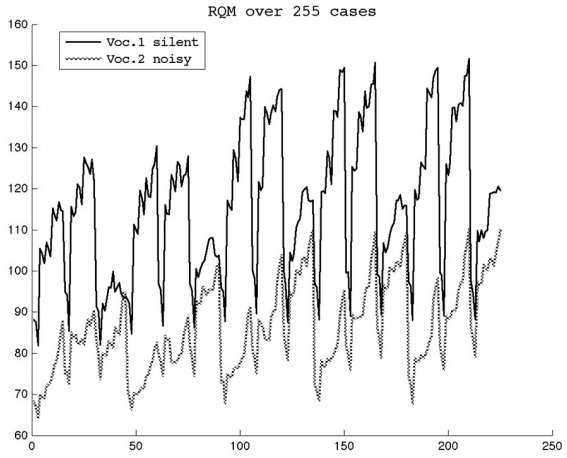


Figure 4: *Robustness Quality Measurement for vocalist 1 silent and noisy dataset over 225 features computation parameters combinations.*

In the performances measurements over the independent signals, the $Q_1$ values show a decrease due to a lower RQM nominal value, as presented in the bottom part of Table 3. Therefore noisy environments lead to performance degradation as expected, but the method shows its capability to reject external sources of noise without an excessive penalization of the overall performances.

## 4. CONCLUSIONS AND FUTURE WORK

We presented a method based on individual vocalists, to compute a set of time and value continuous control signals with high independence and low noise, for particular performance datasets. We run a blind search of the computation parameters to minimizing four quality parameters. Experimental results over different vocalist and performances presented coherent results. Additional experiments with different values of the features rejection thresholds and number of independent computed signals also led to result consistency. In general we showed how feature robustness is dependent for the individual voices, and that the best computation parameters must vary according to the vocalist. The computation of independent signals strongly depends on the specific vocal performance and must be tuned accordingly. Moreover we found and highlighted recurrent pattern in the RQM and ISQM measurements across different features and feature computation parameters. Additional work on alternative techniques to generate independent signals may further improve this method.

The capability of computing noise-free and independent signals from the voice cannot be taken as providing evidence of their "human-controllability". This important HCI issue is still open and we will investigate it in the future. Moreover, since this method can be considered unsupervised, it is necessary to provide the user with information about how the independent signals are extracted from the voice in order to support user control of the system. To address these two issues we developed a real-time application for the independent signal computation, but it needs to cooperate with a system that provides feedback to the user in acoustic and visual forms.

## 5. ACKNOWLEDGMENTS

This work was supported by a scholarship from the NUS Graduate School for Integrative Sciences & Engineering.

## 6. REFERENCES

[1] T. Igarashi and J. F. Hughes, "Voice as sound: using non-verbal voice input for interactive control," in *Proceedings of the 14th annual ACM symposium on User interface software and technology*, 2001, pp. 155–156.

[2] J. A. Bilmes, X. Li, J. Malkin, K. Kilanski, R. Wright, K. Kirchhoff, A. Subramanya, S. Harada, J. A, P. Dowden, and H. Chizeck, "The Vocal Joystick: A Voice-Based Human-Computer Interface for Individuals with Motor Impairments," in *Human Language Technology Conf. and Conf. on Empirical Methods in Natural Language Processing*, Vancouver, Canada, 2005, p. 995–1002.

[3] S. Harada, J. O. Wobbrock, J. Malkin, J. A. Bilmes, and J. A. Landay, "Longitudinal study of people learning to use continuous voice-based cursor control," in *Proceedings of the 27th international conference on Human factors in computing systems*, 2009, pp. 347–356.

[4] S. Harada, J. O. Wobbrock, and J. A. Landay, "Voicedraw: a hands-free voice-driven drawing application for people with motor impairments," in *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, 2007, pp. 27–34.

[5] B. House, J. Malkin, and J. Bilmes, "The VoiceBot: a voice controlled robot arm," in *Proceedings of the 27th international conference on Human factors in computing systems*, 2009, pp. 183–192.

[6] A. Loscos and T. Aussenac, "The wahwactor: a voice controlled wah-wah pedal," in *Proceedings of the 2005 conference on New interfaces for musical expression*, 2005, pp. 172–175.

[7] J. Janer and M. De Boer, "Extending voice-driven synthesis to audio mosaicing," in *5th Sound and Music Computing Conference, Berlin*, 2008, vol. 4.

[8] R. A. Fiebrink, "Real-time Human Interaction with Supervised Learning Algorithms for Music Composition and Performance," Ph.D. Thesis, Princeton, 2011.

[9] S. Fasciani and W. Lonce, "A Voice Interface for Sound Generators: adaptive and automatic mapping of gestures to sound," in *Proceedings of the 2012 conference on New interfaces for musical expression*, 2012.

[10] M. Rocamora and P. Herrera, "Comparing audio descriptors for singing voice detection in music audio files," in *Brazilian Symposium on Computer Music, 11th. San Pablo, Brazil*, 2007, vol. 26, p. 27.

[11] H. Lukashevich, M. Gruhne, and C. Dittmar, "Effective singing voice detection in popular music using arma filtering," in *Workshop on Digital Audio Effects (DAFx'07)*, 2007.

[12] D. Stowell and M. D. Plumbley, "Robustness and independence of voice timbre features under live performance acoustic degradations," in *Proc. of the 11th Int. Conference on Digital Audio Effects*, 2008.

[13] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *J. Acoust. Soc. Am.*, vol. 87, no. 4, pp. 1738–1752, Apr. 1990.

[14] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 578–589, Oct. 1994.

[15] C. Ssnderson and K. K. Paliwal, "Effect of different sampling rates and feature vector sizes on speech recognition performance," in *TENCON'97. IEEE Region 10 Annual Conference. Speech and Image Technologies for Computing and Telecommunications., Proceedings of IEEE*, 1997, vol. 1, pp. 161–164.

[16] B. B. Monson, "High-Frequency energy in singing ans speech," Ph.D. Thesis, University of Arizona, 2011.

[17] P. Comon, "Independent component analysis, a new concept?," *Signal Process.*, vol. 36, no. 3, pp. 287–314, Apr. 1994.

[18] A. Hyvarinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Transactions on Neural Networks*, vol. 10, no. 3, pp. 626–634, May 1999.

[19] G. J. Székely, M. L. Rizzo, and N. K. Bakirov, "Measuring and testing dependence by correlation of distances," *The Annals of Statistics*, vol. 35, no. 6, pp. 2769–2794, Dec. 2007.

[20] A. Gretton, K. Fukumizu, and B. K. Sriperumbudur, "Discussion of: Brownian distance covariance," *The Annals of Applied Statistics*, vol. 3, no. 4, pp. 1285–1294, Dec. 2009.