THE VIBRATO AND PHONEME RATING TOOL: A REAL-TIME SPEECH THERAPY FEEDBACK TOOL

R. de Fréin* and S. Rickard*

* Sparse Signal Processing Group, University College Dublin, Dublin, Ireland

rdefrein@ee.ucd.ie

Abstract: This paper describes the development of a suite of real-time speech therapy tools for vibrato and phonation training. The tools provide real-time visual feedback to the user in the form of a numeric score which can be visualised as a moving two coloured column in the case of the phoneme tool and a rectangle that varies in size both horizontally and vertically in the case of the vibrato tool. The numeric score is calculated by matching parameters extracted from the user's input with those extracted from an ideal vocalization in the case of the phoneme tool and by making measurements on the pitch period of the phrase sung by the user in the case of the vibrato tool. The graphic feedback enables the user to experiment with their vocalization and achieve some specified goal during the practice session. These are proof of concept tools that are intended to serve as a teaching aid that will facilitate practice in-between supervised sessions with a speech or music coach. Trials with these tools indicate that the relationship between the rating of the users vocalization and sound produced by the user is consistent.

Introduction

/s/ and /sh/ differentiation is a classic problem in speech therapy especially for the deaf and patients with cleft palate [1, 2]. The patient can find it difficult to replicate the speech therapists pronunciation, as the tongue is not visible when /s/ and /sh/ are pronounced. The delay between the utterance and feedback is a major disadvantage with batch-processing style tools as the patient cannot be expected to remember the exact sequence in which they made the different vocalization shapes and then compare them with the feedback after the recording session. The success of real-time implementation of speech therapy tools has been reported by [3] "the real-time feedback of essential aspects of production helps in the training of perception and can substantially improve self monitoring."

Vibrato is the slightly wavering quality that a trained singer produces by temporally modulating the pitch of a sustained tone. It has been shown that vibrato can be characterized using measures of vibrato rate and extent [4, 5, 6]. Classically trained singers have great control over both the rate and the extent of their vibrato yet human analysis of vibrato is subjective and hence the rate and extent of vibrato two singers employ for the same German *lied* for example may be quite dissimilar. The reason for the disparity of the two performances lies in the difficulty the student might have with replicating a standard performance of the piece or the fact that different schools of thought suggest a different style of performance. A real-time vibrato feedback system could prove invaluable in the class room as a self tutoring and self awareness tool as it would give unbiased feedback to the user. It could also be used to set guidelines for the rate and extent of vibrato suitable for the piece of music, similar to the guidelines given for the suggested speed, rhythm and style on the musical score.

Development of the Vibrato Tool

Real-time input is obtained using a mono-microphone input and the data acquisition toolbox in matlab. In both tools the signal is sampled at 8*khz* with a window length of 250 samples that is taken using an infinitely repeating trigger every 0.0312 seconds. The window is chosen to be 250 samples long as period estimation using a method such as autocorrelation (AC) [7] or YIN [8] requires that the length of the signal be at least twice the length of the highest pitch period (in this case a period of 125 samples) so that when the first half of the signal overlaps the second half a local maximum will be registered indicating high correlation in the AC method and a local minimum in the case of the YIN algorithm. Many approaches have been developed to evaluate pitch with high resolution e.g. autocorrelation and cepstrum based methods amongst others discussed in [7, 9, 10, 11]. The Yin algorithm was chosen for real-time implementation, as there is no upper limit on its search range.

Pitch Determination

The pitch is determined using the Yin algorithm which is outlined as follows: A windowed input signal x_t, is by definition periodic, if for a time shift of T

$$x_t - x_{t-T} = 0 \tag{1}$$

Therefore, the difference function defined as $d_t(\tau)$ can be used to find the period by searching $d_t(\tau)$ for zeros. The indices corresponding to the zeros are found at multiples of the period of the signal (see Figure 1).

$$d_t(\tau) = \sum_{j=1}^{W} (x_j - x_{j+\tau})^2 \tag{2}$$

ISSN: 1727-1983 © 2005 IFMBE

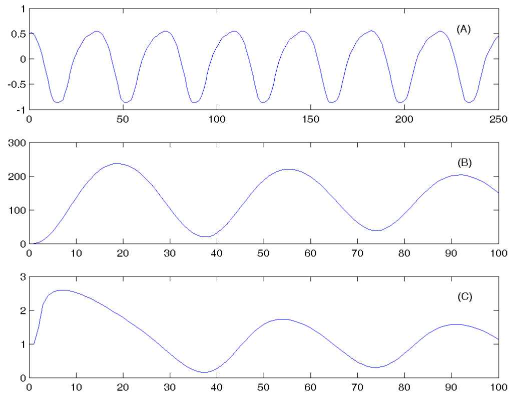

Figure 1: (A) One frame of the input sound, (B) the corresponding difference function vs. lag, (C) the corresponding cumulative mean normalised difference function vs. lag. The period is the index of the first minimum which occurs at lag 37 in plot (B) and (C).

Because of the zero at zero lag in the difference function $d_t(\tau)$, unless a lower limit is set on the search range, the algorithm will choose an erroneous period estimation. The cumulative mean normalized difference function (CMND) is computed as it starts at one and remains quite large for the first few lags, which means that there is no need for a lower search bound.

$$ddash_t(\tau) = \begin{cases} 1, & \text{if}\,,\tau = 0, \\ d_t(\tau) / \left[\frac{1}{\tau} \sum_{j=1}^{\tau} d_\tau(j) \right] & \text{otherwise} \end{cases}$$

$ddash_t(\tau)$ is binary masked with an absolute threshold of 0.7 and the minimum is located. Using parabolic interpolation a second order polynomial is fit to the points in $d_t(\tau)$, that correspond to the lowest minimum in the masked $ddash_t(\tau)$. This increases the resolution of the resulting period estimate. The period estimate is calculated by differentiating the polynomial that has been fit to the masked points of $d_t(\tau)$ and setting it to zero. The index corresponding to the minimum value is the period estimate for this segment of input data. The period estimate is then converted to the corresponding frequency in hertz. For example in Figure 1 the input singing (A), difference function (B) and the CMND (C) are shown. The pitch period is found at the first minimum in the CMND function (C) which corresponds to a lag of 37. Note that the CMND doesn't fall below zero in the first few lags as in the case of the difference function (B) which avoids the problem having to select a lower bound on the search region due to the zero at zero lag.

Vibrato Determination

16 pitch period estimates are calculated per second and are subsequently converted to frequencies in hertz and then used to calculate the rate and extent of vibrato.

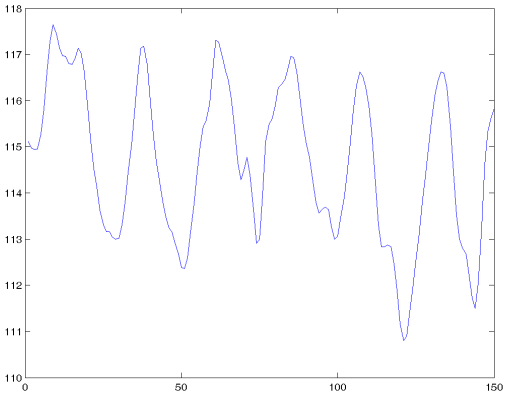

Figure 2: This plot illustrates the interpolated frequency estimates in hertz for a segment of 2.3 seconds of the note $A_2^{\#}$ sung by a male voice with vibrato.

The rate of the vibrato is calculated using the method discussed in [6] every 0.0624 seconds using the past 16 frequency estimates. The lower bound on the number of frequency estimates per second required to estimate the vibrato rate of a human voice was taken to be 64 estimates per second. Due to the computational burden involved in estimating the CMND, a trade-off was made between resolution i.e. number of frequency estimates per second and computational load. It was decided to compute 16 estimations per second and then to resample the data at four times the original rate using low pass interpolation to obtain the requisite signal for the analysis step as shown in Figure 2. The frequency estimates clearly oscillate about a mean frequenct of 115hz. The number of oscillations per second is given as the rate of the vibrato for that second of frequency estimations. The frequency vector f was shifted down to the x-axis by subtracting the mean from each element of the vector and the number of zero-crossings was calculated. The rate is calculated in hertz and can be calculated as follows:

$$f_{new} = ZC(f - \frac{1}{N} \sum_{i=1}^{N} f(i)) \tag{3}$$

where N is the number of frequency estimates and the zero-crossing function is defined as

$$ZC(f_{new}) = \begin{cases} 1 & \text{if}\,, f_{new} \geq 0, \\ -1 & \text{if}\,, f_{new} < 0 \end{cases}$$

The zerocrossing rate is then calculated by applying a difference function to the adjacent elements of f_{new} and assigning a one to differences of two and zero to differences of zero.

$$\text{rate} = \text{zerocrossing}/2/\text{duration} \tag{4}$$

The extent is measured in semi-tones. The semi-tone can be defined as:

$$\text{semitone} = 12 \log_2 \frac{\text{upper}}{\text{lower}} \tag{5}$$

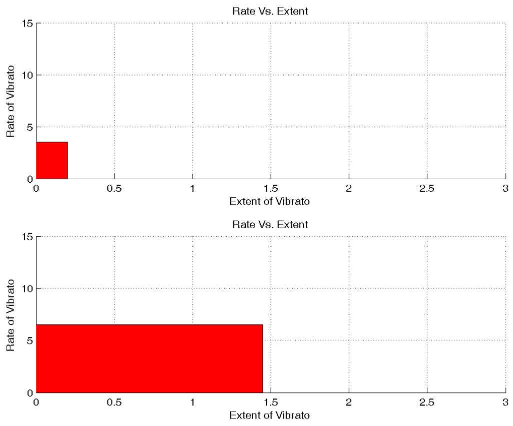

Figure 3: The plot illustrates the rate and extent of vibrato at one instant for the case of a relatively pure tone (in the upper plot) and for the case of a widely spread vibrato (in the lower plot).

where *upper* and *lower* are defined as the absolute extent of f added and subtracted from the mean of f respectively. The absolute extent value is computed using the euclidean norm as follows:

$$\text{absextent} = \frac{||f||_2}{\sqrt{N}}\sqrt{2} \tag{6}$$

Relatively simple graphics were used for the feedback display so that the tool would be intuitive for the first time user. A rectangle was selected as the method of feedback for the vibrato tool as the two degrees of freedom i.e. the horizontal and vertical component could be used to represent both the rate and extent respectively see Figure 3. The output rate and extent were low-pass filtered to give a smoother output display. During the trials of the tool the relationship between the size of the rectangle and the rate and extent of vibrato were quickly established by the user and further experimentation with different forms of feedback led to the conclusion that the rectangle conveyed feedback to the user in the most accessible way.

Development of the Phoneme Tool

Linear prediction (LP) has been considered for spectral estimation in many real-time applications [12, 13, 14]. In the phoneme analysis tool, LP analysis, as discussed in [7, 15] is applied to extract the formants of the input speech and they are then represented in the $F1/F2$ domain where $F1$ and $F2$ correspond to the first and second formant of the speakers vocalization. The formant frequencies are used to characterize phonemes. This tool is based on the supposition that different phonemes can be localized to different regions in the $F1/F2$ space and hence the user's vocalization can be classified by locating its position in the $F1/F2$ domain relative to the position of the centroid of an ideal vocalization. The approach of mapping speech in the $F1/F2$ space has been shown to be successful in [13, 14]. The user's performance is rated using the euclidean distance measures in the $F1/F2$ domain as shown in Figure 4. The centroids for the templates of the /s/ and /sh/ phonemes are shown in Figure 4. These are denoted by a square and a circle for the /s/ and /sh/ sounds respectively. The euclidean distance of the user's vocalization from these two centroids is used to rate the quality of the user's input. Standard representations (taken from the TIMIT database and a group of 10 speakers) of the /s/ and /sh/ phonemes are used as templates for optimum articulation.

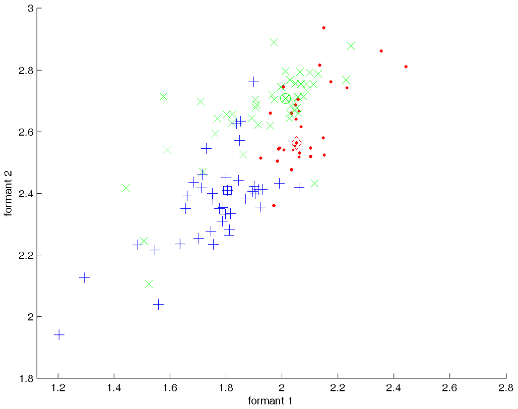

Figure 4: This figure is a plot of the first and second formants of the /s/ and /sh/ templates indicated by '+' and 'x' respectively, their centroids a square and circle respectively and finally it shows the user's vocalization of an attempted /sh/ sound. The centroid is portrayed by a diamond. The euclidean distance was used as a measure of the similarity of the user's vocalization to both the templates.

Formant Extraction

The formant extraction is accomplished as follows: The signal is windowed and the Short-Time Energy Function (STEF) [7] is used to make the voiced/unvoiced binary mask for the windowed segment x with a threshold of 0.05

$$s(x) = \sum_{i=1}^{i=W} x^2(i) \tag{7}$$

where $W = 250$ is the window length. The masked signal is then filtered with a pre-emphasis filter with coefficients $pre = [1 - 0.9]$ to compensate for the $-6dB$ due to a combination of $-12dB/octave$ due to the voiced excitation source and the $+6dB/octave$ as described in [16]. P LP coefficients are calculated using the autocorrelation method as described in [15].

$$\hat{x}_{masked}(n) = \sum_{k=1}^{P} a_k x_{masked}(n-k) \tag{8}$$

Where $P = 8$ is the number of coefficients and equation (8) denotes how the current sample $\hat{x}_{masked}(n)$ is estimated by using a linear combination of the past samples. The AC method minimizes the prediction error $e(n)$ and the vector of coefficients a is obtained by solving

$$\sum_{k=1}^{P} R(i-k) = R(i) \tag{9}$$

where $i = 1, \cdots, p$ and

$$R(i) = \frac{1}{N} \sum_{n=i}^{N-1} u(n)u(n-i) \tag{10}$$

where $u(n)$ is the windowed signal $x_{masked}(n)$ with $n = 0, \cdots, N-1$. The real part of the coefficients a are then converted into the corresponding frequencies (in radians) by calculating the angle of the roots of the polynomial. These frequencies are then sorted by size and the second and the third are chosen as $F1$ and $F2$. The first frequency is rejected as it is usually due to the pitch of the word being sung by the speaker.

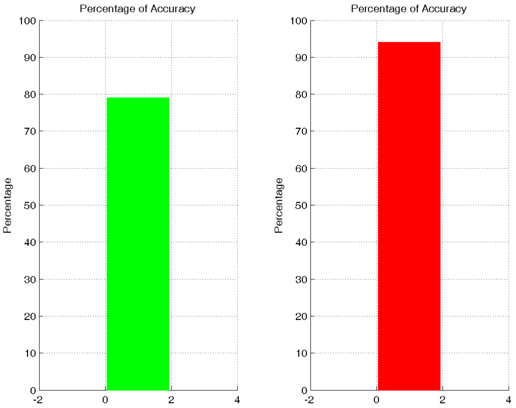

Figure 5: The is a frame of the realtime feedback from the vibrato tool. The column on the left indicates the user's vocalization of a /sh/ sound with a score of 79% and the red column on the right indicates the score of the user's vocalization of a /s/ sound 95%.

The phoneme tool gives feedback to the user using a single column with varying height (See Figure 5). The colour of the column indicates the phoneme being vocalized at that instant and the height of the column indicates the score assigned to the user's vocalization. The score given as feedback to the user is calculated by calculating the euclidean distance of the centroid of both of the templates from the user's vocalization as shown in Figure 4. These distances are then normalized and the /s/ /sh/ decision is made by choosing the smallest distance and the colour of the column assigned based on that decision. The percentage score is the ratio of smallest distance in the $F1/F2$ space from the user's point to the centroids to the sum of both of the distances from the point to the centroids. This percentage is then subtracted from 100% and displayed as the height of the column see Figure 5. The output percentage is averaged over the past 5 readings to give a smoother output display. A black bar indicates that the input sound was not a /s/ or /sh/ sound.

Results

The vibrato analysis tool was tested with a group of three male (one bass and two tenors) and three female (one alto and two mezzo sopranos) classically trained singers. All of them reported a high correlation between the perceptual degree of vibrato of the their vocalization and the assigned rating in the tests. The program was run for 30 seconds for each test.

Table 1: Vibrato Test Results I

Singer	Max Rate (hz)	Max Ext. (semitones)
Bass1	5.38	1.35
Tenor1	5.67	1.30
Tenor2	5.57	1.60
Alto1	5.05	1.68
Mezzo Soprano1	5.37	0.68
Mezzo Soprano2	4.38	1.38

Table 2: Vibrato Test Results II

Singer	Min Rate (hz)	Min Ext. (semitones)
Bass1	0.83	0.32
Tenor1	0.46	0.35
Tenor2	0.51	0.28
Alto1	0.52	0.40
Mezzo Soprano1	1.10	0.37
Mezzo Soprano2	0.50	0.29

Each singer was asked to produce an unwavering note that was comfortably in their range and then to increase the extent of their vibrato to about 1.5 semitones with a rate of about $6hz$. All of the singers found that an extent of a semitone and a half was easy to achieve yet those with little natural vibrato found it difficult to sustain that level of extent when increasing the rate. See Table 1 and Table 2 for a summary of the results. The measurements minimum rate and extent corresponds to the singer's initial unwavering tone. The maximum rate and extent corresponds to the maximum vibrato achieved by each singer in the thirty second period. Each singer was given time to experiment with the apparatus so as to get accustomed to the method of feedback.

The same group of 6 test subjects was used to test the phoneme analysis tool. During trials, the software was run for 30 seconds and the user was asked reproduce a pre-recorded /s/ and /sh/ sound repeatedly. The highest scores for each user was then tabulated see Table 1. In general the users failed to obtain as high as score as the /sh/ score while producing a /s/ sound. This can be attributed to the fact that a relatively small number of subjects was used to create the templates and that some of

Table 3: Phoneme Test Results

Speaker	Max /s/ %	Max /sh/ %
Speaker1	76.99	99.13
Speaker2	74.59	99.38
Speaker3	72.34	99.57
Speaker4	80.75	74.58
Speaker5	76.27	94.01
Speaker6	68.45	82.52

the test subjects had differnet accents which plays a big role in phoneme classification as the second formant $F2$ is greatly affected by the accent of the speaker. Any vocalization with a score of 30% or more was judged to have been distinguishable as a /s/ or a /sh/ vocalization.

Discussion

Two systems have been devised that give the user real-time visual rating on firstly the quality of their vibrato and secondly the quality of their vocalizations of the /s/ and /sh/ phonemes. The user interface was judged to have been both intuitive and easy to use. It became obvious from watching the users that the real-time feedback was engaging and that the challenge of bettering their previous score was an incentive to continue using the tool. Many of the test subjects reported an increased awareness of the mechanics involved in producing both sounds i.e. position of the tongue, jaw and muscles and in turn this increased self awareness improved their performance.

Conclusions

The tools described in this paper succeeded in providing real-time feedback in an intuitive way. Both tools could easily be incorporated into computer games which would make them more appealing to speech impaired children or younger singers [10, 17]. Further work on the implementation of these tools could be put done in reducing the computational load of the YIN algorithm and hence, increase the number of pitch period estimations per second. An increase of the database of templates for the phoneme tool could also lead to better rating of the user's vocalizations and accent independence as an ensemble of points could be used as a centroid for the phoneme template. Further test will be completed with the aid of speech impaired patients and a wider group of professional classically trained singers. There is great scope for these tools in the areas of speech therapy and music tutoring.

References

[1] A. W. KUMMER. Cleft lip/palate and velopharyngeal dysfunction (vpd): The effects on speech and resonance. Cincinnati Children's Hospital Medical Center, Speech Pathology Department.

[2] L. PENA. Cleft lip and palate. Chicago Pediatrics Clinic Curriculum.

[3] E. ABBERTON AND X. HU AND A. FOURCIN. Real-time speech pattern element displays for interactive therapy. *Internation Journal of language and Communication Disorders Vol.33*, pages 292–297, 1998.

[4] I. ARROABARREN AND M. ZIVANOVIC AND J. BRETOS AND A. EZCURRA AND A. CARLOSENA. Measurement of vibrato in lyric singers. *Proceedings of the 18th IEEE Instrumentation and Measurement Technology Conference, 2001,Vol. 3*, pages 1529–1534, 2001.

[5] J. BRETOS AND J. SUNDBERG. Measurement of vibrato parameters in long sustained crescendo notes as sung by ten sopranos. *Speech, Music and Hearing (TMH),Quarterly Progress and Status Report, KTH, Vol 43*, pages 660 – 665, 2002.

[6] E.K. CROWLEY. A vibrato analysis tool. Thesis for Master of Science in Music Technology, UNiversity of Limerick, 2004.

[7] L.R. RABINER AND R.W. SCHAFER. *Digital Processing of Speech Signals*. Englewood Cliffs: Prentice Hall, 1978.

[8] A. DE CHEVEIGNE AND H. KAWAHARA. Yin, a fundamental frequency estimator for speech and music. *ASA Spring Meeting01*, page 1917 1930., 2001.

[9] S. SOOD AND A. KRISHNAMURTHY. A robust on-the-fly pitch esimation algorithm. *MultiMedia'04, October 10-16*, 2004.

[10] P. MCLEOD AND G. WYVILL. Visualization of music pitch. *Proceedings of the Computer Graphics International(CGI'03)*, 2003.

[11] D. GERHARD. Pitch extraction and fundamental frequency: Histroy and current techniques. *Technical report TR-CS,Nov03*, 2003.

[12] F. KEILER AND D. ARFIB AND U. Z ÖLER. Efficient linear precidtion for digital audio effects. *Proceeding of the COST G-6 Conference on Digital Audio Effects (DAFX-00), Verona, Italy, DEC 7-9*, page 1917 1930., 1-6.

[13] Stephen A. Zahorian. Color display of vowels as a speech articulation training aid. *Engineering in Medicine and Biology Society, 1988. Proceedings of the Annual International Conference of the IEEE , 4-7 Nov. 1988*, pages 1530 – 1540, 1988.

[14] Stephen A. Zahorian and Subhashri Venkat. Vowel articulation training aid for the deaf. *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on , 3-6 April 1990*, pages 1121 – 1124, 1990. CH2847-2/90/0000-1121.

[15] J. MAKHOUL. Linear prediction: a tutorial review. *Proceeding of the IEEE,Vol. 63, N0.4*, pages 561–579, 1975.

[16] F.J. OWENS. *Signal Processing of Speech*. The Macmillan Press LTD, London, 1993.

[17] Perttu Hämäläinen and Teemu Mäki-Patola and Ville Pulkki and Matti Airas. Musical computer games played by singing. *Proceedings of the 7th International Conference on Digital Audio Effects (DAFx'04), Naples, Italy, Oct 5-8*, pages 367 – 371, 2004.