Specifying Sounding Frequency of a Voice Model during Live Interactive Saxophone Performance

Jennifer Hsu
University of California, San Diego
jsh008@ucsd.edu

Tamara Smyth
University of California, San Diego
trsmyth@ucsd.edu

ABSTRACT

In this research, a system is developed by which a parametric, physics-based synthesis model of the voice may be controlled by the sound of a saxophone played during real-time performance. Use of physical models in such musical contexts presents challenges due, in part, to the sometimes musically unintuitive (and nonlinear) nature of their physical synthesis parameters. In the voice model here for example, an increase in input pressure from the lungs might (intuitively) increase the amplitude of the produced sound, but (less intuitively and perhaps less desirably) also cause a shift in sounding frequency. Here, we thus address the problem of mapping the higher-level musical parameter of pitch to lower-level physical synthesis parameters, subglottal (lung) pressure and vocal fold resonant frequency. A table matches all combinations of pressure and vocal fold frequency values in their playable range to the corresponding fundamental frequency of the model's produced sound. Curves fit to the table data ultimately yield expressions for either physical parameter as a function of both desired pitch and the other parameter in the pair. Effectiveness of this solution is explored by extracting frequency trajectories from live saxophone input and using them to control the pitch of the voice model.

1. INTRODUCTION

The work presented here is part of a larger project that aims to facilitate the interaction between a saxophonist and a physics-based synthesis model of the human voice during live performance. A step toward having performer and synthesizer seemingly engaged in a sort of duet, where one musical idea or action is a reaction of another, is to first recognize some aspect or feature of the saxophone's sound, and then use it in some way to control a feature of the synthesized voice.

Systems have been developed to detect pitch in live instrument signals [1] and many works use detected pitches in the generation of new audio material [2, 3]. Systems that extract timbre-related features have also been developed, such as [4] which focuses on music gesture recognition in live performance, using found gestures with a genetic algorithm for improvisation with a performer. Other work involves a system that extracts pitch, loudness, and timbre contours from saxophone gestures to guide expressive improvisation from virtual instruments [5].

In this work, the focus is on remapping extracted features to parameters of a physical model and only the most basic of musical features is addressed. Fundamental frequency trajectories are extracted from musical saxophone *gestures*—unified, expressive sonic events as described in [6]—using sinusoidal modeling techniques [7, 8]. Because of the harmonic nature of the signals produced by the saxophone, the computational expense of estimating discrete pitches was unnecessary and, in fact, less effective in preserving the frequency contour of the saxophone gesture.

The extracted frequency trajectory is then used to control the sounding frequency of the voice model—a task made difficult by the fact that nonlinearities in the voice model cause a non-intuitive mapping between its physical parameters and desired musical parameters such as pitch, dynamics, or timbre. Setting the vocal fold valve frequency and/or lung pressure to achieve a particular sounding frequency, is nontrivial. Though this known problem has been discussed in [9, 10] among others, the work presented here proposes a solution that is suitable for real-time live performance.

An overview of the voice model implementation is provided, followed by a method for determining the physical parameter values for the voice model given a desired sounding frequency. Finally, results are given from a practical example in which the voice model imitates a frequency gesture produced by the saxophone.

2. THE VOICE MODEL

The voice synthesis model consists of a dynamic pressure-controlled valve modeling the vibrating vocal folds. This valve is coupled to a piecewise cylindrical one-dimensional digital waveguide, with varying cross-sectional area along its length, modeling the vocal tract shape when producing different vowel sounds.

The generalized valve model was first introduced in [11], providing a configurable model of a pressure controlled valve that allowed for different valve types (blown open, blown closed, and "swinging") suitable for reed instruments and vocal systems, to be implemented simply by setting the model's parameters. Here we use the generalized valve in its blown open configuration to model the oscillating vocal folds and the flow through the valve channel which widens and narrows in accordance with the vocal folds. The volume flow, when multiplied by the characteristic impedance of the vocal tract to which the valve is coupled, provides input pressure to the waveguide model. That input pressure, along with existing pressure reflected from the mouth opening and various other changes in impedance along the vocal tract length, constitute the oscillating pressure at the base of the vocal tract p—an oscillation having a fundamental frequency at what is termed here, the "sounding frequency".

The problem from the view point of parametric control is that the pressure-controlled valve produces a volume flow, and thus an input pressure to the bore (a function of its cross-sectional area) that is nonlinear. The resonant (natural) frequency of the valve can be seen by the equation governing its displacement,

$$\frac{d^2x(t)}{dt^2} + 2\gamma\frac{dx(t)}{dt} + \omega_0^2(x(t) - x_0) = \frac{F(t)}{m}, \qquad (1)$$

where m is the valve mass, γ is the damping coefficient, $\omega_0 = \sqrt{k/m}$ is the resonant frequency (rad/s) of the *ideal* mass-spring system with stiffness coefficient k, with the actual *damped* resonant frequency being slightly lower and given by

$$\omega_v = \sqrt{\omega_0^2 - \gamma^2} \qquad (2)$$

The actual fundamental frequency of vibration, and thus the *sounding* frequency of the model output however, is highly dependent on the driving force $F(t)$—a nonlinear function that is dependent on subglottal pressure p_0, the amplitude of the existing pressure p propagating in the vocal tract, as well as volume flow—all of which see part of the valve's surface area. Furthermore, the vocal tract itself has resonant frequencies (visible by peaks in its input impedance curve), and the pressure that drives its frequency modes, also a function of a nonlinear volume flow, further influences the *sounding* frequency.

Assuming a vocal tract with unchanging length and cross section for a given vowel sound (and thus static resonant frequencies), the main physical control parameters of the voice synthesis model influencing sounding frequency are the subglottal pressure p_0 and the valve resonant frequency,

$$f_v = \frac{\omega_v}{2\pi}\ \text{Hz}, \qquad (3)$$

a change of which is akin to controlling tension in the vocal folds. In the following, a method is suggested for mapping these physical parameters to the more meaningful musical parameter of sounding frequency f_0.

3. SPECIFYING SOUNDING FREQUENCY FOR THE VOICE MODEL

The input parameters to the voice model are subglottal pressure p_0 and valve frequency f_v. Because valve frequency is not a musically intuitive control parameter, a more appropriate input control parameter is the sounding frequency f_0, the fundamental frequency of the produced sound. Though f_0 is certainly related to valve frequency, and indeed they have been found on occasion to be quite close, the valve frequency is lower—as expected of a *blown-open* valve type [9], and also changes as a function of subglottal pressure. In this section, a procedure is proposed for mapping f_0 to ω_v for a specific p_0.

3.1 Sounding Frequency f_0 as a Function of p_0 and ω_v

A table holds fundamental frequency values of the sound produced by the voice model with all combinations of values of subglottal pressure p_0 (in a range of 200-2000 Pa) and valve frequency (in a range of 145-830 Hz). An implementation of McAulay-Quatieri (MQ) analysis [12] is used to determine the fundamental frequency of each synthesized signal.

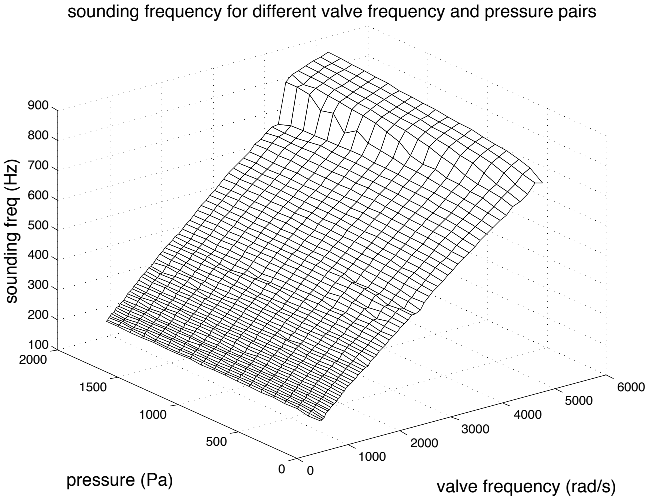

Figure 1. Fundamental frequency of the sound produced by the voice model for each valve frequency (145-830 Hz) and subglottal pressure (200-2000 Pa) pair. Lines are fit across valve frequency values for each pressure curve.

Figure 1 shows a plot of the sounding frequency as a function of subglottal pressure and valve natural frequency as given by (1). The relationship between valve frequency and sounding frequency is practically linear, but there is slight variation in the slope across different subglottal pressure values. A line is thus fit to each of the curves relating sounding frequency f_0 and valve frequency ω_v for each value of subglottal pressure p_0:

$$f_0(\omega_v, p_0) = b_{0,p_0} + b_{1,p_0}\omega_v, \qquad (4)$$

where b_{i,p_0} are the coefficients found by the linear fit for pressure value p_0.

3.2 Finding Valve Frequency Given a Sounding Frequency and Subglottal Pressure

Given that two parameters affect the sounding frequencies, the simplest solution is to keep one fixed while perturbing the other. A more elaborate strategy could be developed whereby, for instance, one takes the combined shortest distance to *both* parameters that produce the desired frequency, potentially changing both parameters to “minimize effort”. Satisfactory results have been achieved using the following approach.

Given a desired sounding frequency, f_0, and a subglottal pressure p_0, we consider two neighboring pressure curves having the closest pressure value to p_0: p_-, the closest value below and p_+, the closest value above p_0.

Curve coefficients b_{i,p_-} and b_{i,p_+} are then linearly interpolated to produce b_{i,p_0} and used in a rearrangement of (4) to calculate an estimated valve frequency

$$\tilde{\omega}_v = \frac{f_0 - b_{0,p_0}}{b_{1,p_0}} \quad (5)$$

for desired sounding frequency f_0.

Lastly, $\tilde{\omega}_v$ and the input subglottal pressure p_0 is fed into the voice model. Because interpolated pressure values p_0 are used in the calculation of (5), the estimated value for $\tilde{\omega}_v$ will not necessarily result in the *exact* desired sounding frequency when used in the model. The next section provides an evaluation on the accuracy of the f_0 specification method.

3.3 Evaluation of Table Accuracy

The accuracy of the calculated valve frequency is limited by the pressure values used to build the tables and generate curves. From preliminary listening tests, results are satisfactory. A quantitative measure of the difference between the table output and that produced by the model set with the same parameter values is useful to get a better idea of the method’s accuracy.

In particular, to verify the accuracy of the coefficient interpolation, a test set of desired sounding frequencies and subglottal pressures is created such that none of the values is actually present in the table. The test set is then used as input to the voice model to synthesize a collection of output signals from which the fundamental frequency is determined. The error is measured as the absolute difference between the desired sounding frequency and the true sounding frequency. The mean error is 11.58 Hz with a standard deviation of 10.91 Hz with a maximum error of 91.58 Hz and a minimum error of $1.2e^{-2}$ Hz. The accuracy is higher in sounding frequencies produced at mid-to-lower frequencies and at middle values for subglottal pressure. The voice model was designed to simulate the physical properties of the human voice, so performing less well at the extreme pressure and frequency values is expected.

4. EXTRACTING FREQUENCY TRACKS FROM SAXOPHONE PERFORMANCE

A goal of this project is to control the voice model with features derived from the acoustic saxophone signal. The synthesized voice can be designed to either imitate or contrast in response to the saxophone gesture. Here, we focus on extracting the first harmonic of the saxophone signal (found in almost all cases to be the fundamental) so that it can be mapped to the sounding frequency of the voice model. Initial experiments with pitch trackers did not give satisfactory results as most pitch trackers report discrete pitches without taking into account a changing frequency trajectory. Rather, what is desired is a frequency trajectory from the saxophone signal that matches our perception of how the frequency changes over the gesture. To accomplish this task, we use MQ analysis to find frequency *tracks*, time-varying partials, and choose the most likely frequency trajectory out of the frequency tracks.

4.1 Determining Frequency Tracks

The signal is first analyzed using a short-time Fourier transform (STFT) with a Hanning window of size 1024 samples and 50% hopsize. The spectrogram of the signal is analyzed using an MQ implementation in Matlab [12]. This implementation picks out peaks from each frame of the magnitude spectrum and arranges these peaks into frequency tracks that continue along the frames.

4.2 From Frequency Tracks to a Frequency Trajectory

From the frequency tracks output by the MQ analysis, the fundamental frequency of the saxophone signal is captured by the frequency track with the lowest frequency. If the current frequency track ends, the frequency trajectory continues by switching to the lowest frequency track, if one exists. If a lower frequency track exists, the frequency trajectory continues by switching to that lower frequency track. This results in a frequency trajectory that matches the trajectory of the fundamental frequency of the saxophone signal. In the upper portion of Figure 2, the spectrogram of a saxophone gesture with the frequency tracks overlaid is plotted. The desired frequency trajectory is outlined in white. The bottom portion of the figure is the extracted fundamental frequency trajectory.

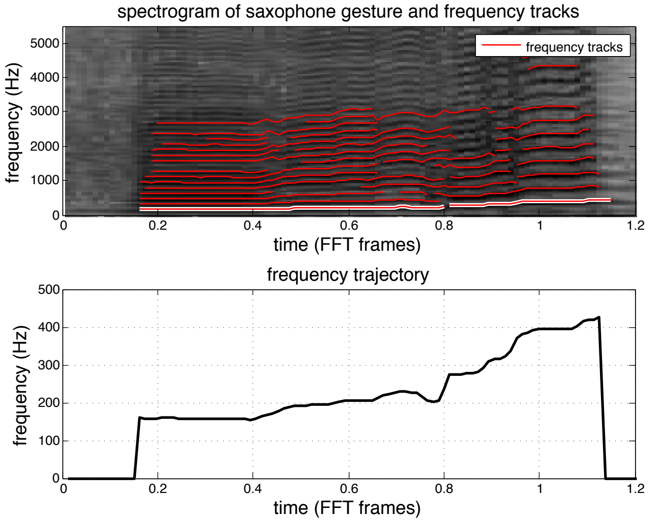

Figure 2. Frequency tracks found in a saxophone gesture are overlaid on the magnitude spectrogram of the saxophone signal. The desired frequency tracks are outlined in white. Below is the extracted frequency trajectory.

These frequency trajectories can be input to the voice model as the desired sounding frequency to mimic the saxophone sound. To contrast with the saxophone, different transformations can be made on the frequency trajectories before feeding them into the voice model.

The resulting frequency trajectory from Figure 2 was used as the sounding frequency input to the voice model to create the plot shown in Figure 3. The synthesized voice signal is run through the frequency trajectory algorithm to verify whether or not the output frequency trajectory matches the input. The upper plot in Figure 3 shows the magnitude spectrogram of the voice signal with frequency tracks overlaid in red. The frequency track outlined in white is the desired frequency trajectory. The bottom plot in Figure 3 is the resulting frequency trajectory from the voice signal. This frequency trajectory matches the input trajectory as shown in Figure 2.

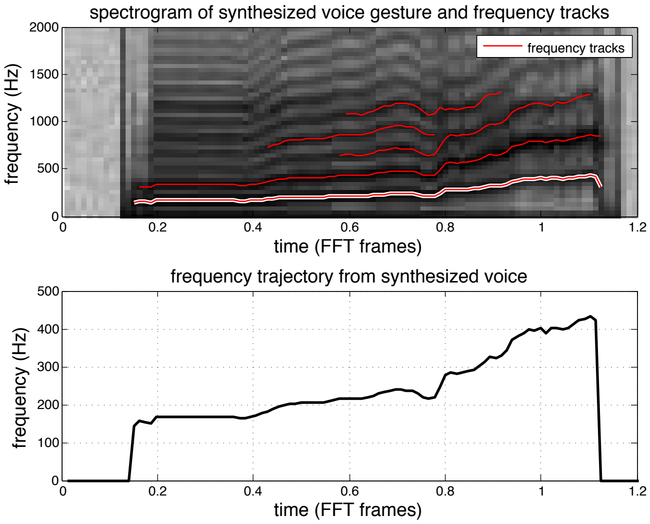

Figure 3. Frequency tracks, with desired track outlined in white, overlaid on the magnitude spectrum of a synthesized voice signal driven by the frequency trajectory from Figure 2. The lower plot is the resulting trajectory of the voice signal.

5. CONCLUSION

This research has two objectives: 1) to refine a physical voice model for intuitive control and 2) to extract expressive contours from saxophone gestures to be used as input to the voice model. We presented a method for determining sounding frequency for a physical voice model and gave an example of extracting a frequency trajectory from a saxophone signal to be used as input to the voice model. Controlling the parameters of the voice model with other sonic elements extracted from the saxophone signal is left for future work.

Acknowledgments

We would like to express our gratitude to the Canada Council for the Arts (CCA) for their generous funding through a collaboration with saxophonist and composer Joel Miller. We are also grateful to the Center for Research in Entertainment and Learning (CREL) who, though support from the Office of Graduate Studies, provide research opportunities to graduate students in Music at UCSD.

6. REFERENCES

[1] M. Puckette, T. Apel, and D. Zicarelli, "Real-time audio analysis tools for Pd and MSP," in *Proceedings of the International Computer Music Conference*, San Francisco, 1998, pp. 109–112.

[2] C. Lippe, "A composition for clarinet and real-time signal processing: Using Max on the IRCAM signal processing workstation," in *Proceedings of the 10th Italian Colloquium on Computer Music*, 1993, pp. 428–432.

[3] G. Lewis, "Too many notes: Computers, complexity, and culture in Voyage," *Leonardo Music Journal*, vol. 10, pp. 33–39, 2000.

[4] D. Nort, "A system for musical improvisation combining sonic gesture recognition and genetic algorithms," in *Proceedings of the Sound and Music Computing Conference*, Porto, Portugal, 2009, pp. 131–136.

[5] W. Hsu, "Managing gesture and timbre for analysis and instrumement control in an interactive environment," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, Paris, France, 2006, pp. 376–379.

[6] O. Ben-Tal, "Characterising musical gestures," *Musicae Scientiae*, vol. 16, no. 3, pp. 247–261, 2012.

[7] X. Serra, "A system for sound analysis/transformation/synthesis based on a deterministic plus stochastic decomposition," Ph.D. dissertation, Stanford University, 1989.

[8] R. McAulay and T. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, no. 4, pp. 744–754, Aug 1986.

[9] N. H. Fletcher and T. D. Rossing, *The Physics of Musical Instruments*. Springer-Verlag, 1995.

[10] J. Gilbert and J. Kergomard, "Calculation of the steady-state oscillations of a clarinet using the harmonic balance technique," *J. Acoust. Soc. Am.*, vol. 86, no. 1, pp. 35–41, 1989.

[11] T. Smyth, J. Abel, and J. Smith, "A generalized parametric reed model for virtual musical instruments," in *Proceedings of the International Computer Music Conference*, Barcelona, Spain, 2005, pp. 347–350.

[12] D. P. W. Ellis, "Sinewave and sinusoid+noise analysis/synthesis in Matlab," 2003, online web resource. [Online]. Available: http://www.ee.columbia.edu/~dpwe/resources/matlab/sinemodel/