Virtual Flute

REALSIMPLE Project*
Edgar J. Berdahl and Julius O. Smith III
Center for Computer Research in Music and Acoustics (CCRMA), and the
Department of Electrical Engineering
Stanford University
Stanford, CA

Abstract

This laboratory assignment should teach students how to model the physics of a flute using a digital waveguide. After explaining the acoustics of a tube with two open ends, we develop a simple digital waveguide flute model. Next, the model is extended to include a noise excitation source and a nonlinear element. Students investigate the behavior of the flute model in `pd` and answer some questions to solidify understanding. The prerequisite assignments are the monochord laboratory assignment,[1] the weighted monochord laboratory assignment,[2] the harmonics laboratory assignment,[3] the traveling waves laboratory assignment,[4] the digital waveguide model laboratory assignment,[5] and the virtual acoustic tube lab.[6]

Contents

*Work supported by the Wallenberg Global Learning Network
[1]http://ccrma.stanford.edu/realsimple/lab_inst/
[2]http://ccrma.stanford.edu/realsimple/weighted_mono/
[3]http://ccrma.stanford.edu/realsimple/harmonics/
[4]http://ccrma.stanford.edu/realsimple/travelingwaves/
[5]http://ccrma.stanford.edu/realsimple/waveguideintro/
[6]http://ccrma.stanford.edu/realsimple/vir_tube/

1 Summary Of Objectives

- Explain the acoustics of a *tube with two open ends*.
- Discuss the basic operation of a *digital waveguide flute model*.
- Extend the model to include a *noise excitation source* and a *nonlinear element*.
- Allows students to investigate the behavior of the *flute model in `pd`*.
- Ask students to build on what they have learned to *derive the signal flow diagram* and answer other questions about a woodwind instrument with one open end and one closed end.

2 Traveling Waves In A Tube With Open Ends

Longitudinal waves propagating in a tube can be modeled approximately by a one dimensional waveguide. The standard wave variable used for analysis is the pressure $p(x,t)$ at any point x along the tube and time t. Recall from the traveling waves laboratory assignment[7] that the wave variable $p(x,t)$ can be decomposed into left-going $p_l(x,t)$ and right-going $p_r(x,t)$ traveling wave components.

$$p(x,t) = p_l(x,t) + p_r(x,t) \tag{1}$$

From the virtual acoustic tube lab[8], we know that when two cylindrical tubes are joined together, the discontinuity will cause traveling wave components to reflect according to the reflectance k. A traveling wave leaving a tube with radius r_1 and entering a tube with radius r_2 will reflect back into the tube with radius r_1 according to

$$k = \frac{r_1^2 - r_2^2}{r_1^2 + r_2^2}. \tag{2}$$

Contemplate what traveling pressure wave components might do at the open end of a tube. We could approximate the region outside of the tube as being a tube with infinite radius. To find the reflectance, we can write

$$k_{open} = \lim_{r_2 \to \infty} \frac{r_1^2 - r_2^2}{r_1^2 + r_2^2} = \lim_{r_2 \to \infty} \frac{-r_2^2}{r_2^2} = -1 \tag{3}$$

$k_{open} = -1$ means that pressure waves reflect approximately with a simple sign inversion from an open end. This means that the reflecting wave will cancel the impinging wave at the open end, creating a node. For more intuition on standing waves in wind instruments, see some depictions of standing waves on the web;[9] however, to avoid confusion, make sure you understand that *pressure* waves have nodes where *displacement* waves have anti-nodes and vice versa.

[7] http://ccrma.stanford.edu/realsimple/travelingwaves/
[8] http://ccrma.stanford.edu/realsimple/vir_tube/
[9] http://cnx.org/content/m12589/latest/

3 Traveling Waves In A Flute

Here we develop a very simplified digital waveguide model of waves propagating in a flute. The far end of the flute from the player's mouth has an open end, so to first approximation, pressure waves reflect with a sign inversion from the far end. Flutists can shorten the effective length of the tube by opening holes along the length of the tube. The effective length then corresponds to the first open hole. In contrast with the clarinet, saxophone, etc., the end of the flute near the player's mouth, which is known as the head, behaves acoustically more like an open end than a closed end [3]. This is because a flute player only places the lower lip against the embouchure hole—he or she does NOT completely cover the hole. Consequently, pressure waves reflect with a sign inversion at the head of the flute.

The simplified model is shown in Figure 1. We use the same basic structure as with the vibrating string in the digital waveguide model laboratory assignment,[10] although this is somewhat coincidental because here we are modeling sound pressure waves rather than structural displacement waves. The total delay of N samples around the loop corresponds to the note being played. Since both terminations support inverting reflections, the fundamental frequency $f_0 = \frac{1}{NT}$ where T is the digital sampling interval in seconds. To make sure that waves circulating in the waveguide decay over time, we change one of the gains from -1 to $-g$ where $g \approx 1$ but $g < 1$. So that the higher harmonics decay more quickly than the lower harmonics, we insert a lowpass filter into the feedback loop (see Figure 1) [4].

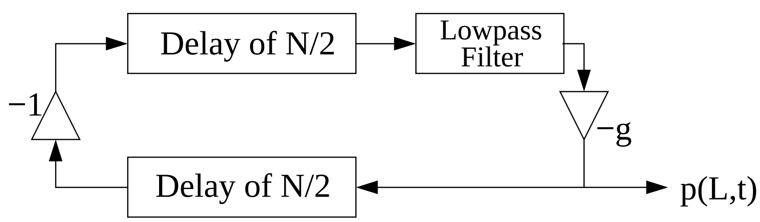

Figure 1: Very simplified digital waveguide model of pressure waves propagating in a flute

4 Noise Excitation Source

We need to augment the model to explain how the instrument is excited. Flute players blow a jet of air against a sharp edge, which is placed on the boundary of the pipe. The jet is unstable and tends to flow on only one side of the edge. However, since the attached tube tends to resonate only at harmonics of the fundamental frequency of the tube, the tube causes the jet to switch back and forth between flowing on either side of the edge. This switching occurs at approximately the fundamental frequency of the note being played.

The switching of the jet causes the end of the jet to break up into vortices and turbulence. This nonlinear effect causes noise pressure waves to be injected into the tube [2]. This noise *excites* the tube can be included in the model as shown in Figure 2. The degree of noise excitation is related to how hard the flutist is blowing. Note that this noise is *broadband* since it contains energy at all of the frequencies in the audible spectrum.

[10] http://ccrma.stanford.edu/realsimple/waveguideintro/

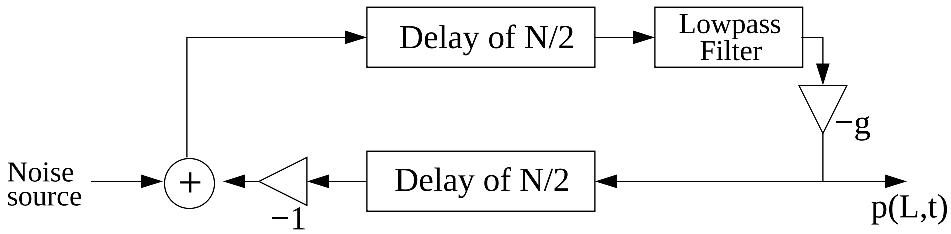

Figure 2: Simplified flute model including a noise excitation source

5 Full Model

The nonlinear characteristic of the jet implies also that pressure waves impinging on the head termination actually reflect in a nonlinear fashion. We can adjust the model as shown in Figure 3 by inserting the nonlinear element NL into the loop. NL can be approximated using something similar to the soft-clipper combined with an extra delay line and an additional reflection [1]. In simplified terms, NL essentially means that incoming pressure waves with large magnitudes are reflected with gains closer to 0 than -1.

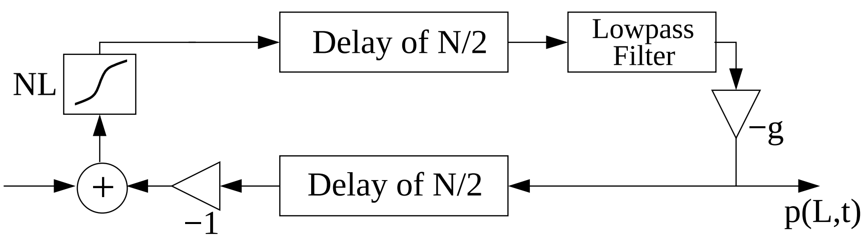

Figure 3: Digital waveguide model of a flute

6 Investigation In `pd`

Now that you understand the basics of the virtual flute model, you will have the opportunity to "play" it in `pd`.

1. Install the `flute~` extern.
   - If you are running Linux on an Intel-architecture processor, then download the extern `flute~.pd_linux`[11]. Place it in a directory with your other externs. For instance, this might be `~/externs`. Then make sure that this directory is included in the path for `pd`. You can do this by adding `-path [PATHNAME]` to the end of the `pd` command when you invoke it at the command line interface. Alternatively, you may add `-path [PATHNAME]` to the file `~/.pdrc`.
   - If you are not running Linux on an Intel-architecture processor, then you will need to recompile the `flute~` extern. We have provided a package[12] for doing this, although

[11] http://ccrma.stanford.edu/realsimple/vir_flute/flute~.pd_linux
[12] http://ccrma.stanford.edu/realsimple/vir_flute/compile-flute.zip

you may also want to consult the general instructions in the tutorial on embedding STK instruments in Pure Data externals.[13]

2. Download the `pd` patch `flute.pd`[14], and open it in `pd`.

3. Ensure that the patch is not in editing mode, check the "compute audio" box in the main `pd` window, and increase the "Output volume" slider until the the volume level is comfortable.

4. To adjust values stored in number boxes, you can either click on a box, type in a new number, and press *enter*, or you can drag up and down from the box with the mouse. Using one of these methods, increase the "Frequency" number box to 400. Now the spectrum should look similar to the one shown in Figure 4.

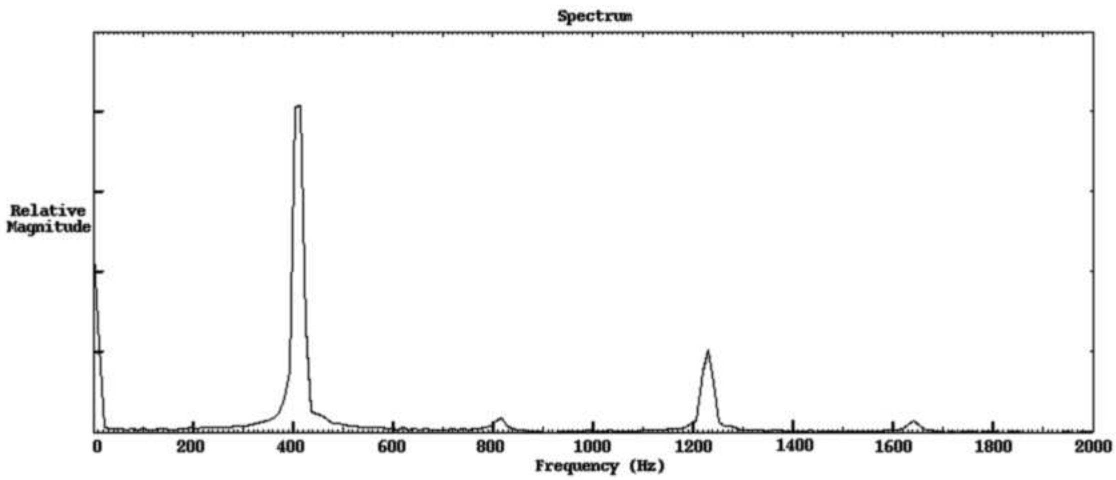

Figure 4: Example spectrum of `flute~` output with fundamental frequency slightly more than 400Hz

5. If the spectrum looks too small, it may help to also increase the "Amplitude" number box. The amplitude adjustment is responsible both for the average pressure in the flute due to the flutist's breath as well as a scaling of the output signal $p(L, t)$.

6. Consider the difference in behavior if you blow softly on a real flute, if you blow normally on a real flute, or if you blow very hard. Carry out this experiment on the virtual model with the "Breath Pressure" slider. First move the slider all of the way to the right, which corresponds to blowing hard. How does the model behave? Now investigate the behavior for intermediate and low breath pressures. Is this what you expected?

7. Now try toggling the "Blowing On/Off" switch on and off. When the toggle is off, no breath pressure is applied. When the toggle is on, breath pressure is applied according to the "Breath Pressure" slider. Listen carefully to the sound of the model when the toggle is switched. This should sound like a real flute. This is one of the advantages of physical models. Because the computer is simulating physical behavior, changing model parameters in real time results in especially realistic sounds.

[13] http://ccrma.stanford.edu/realsimple/stkforpd/

[14] http://ccrma.stanford.edu/realsimple/vir_flute/flute.pd

8. After adjusting parameters for a while, you may find that the pitch produced by `flute~` differs significantly from the number in the "Frequency" number box. When this happens, the pitch will generally differ by an octave or two. This means that the virtual flute player is overblowing the virtual flute. (You can tell approximately what pitch is being produced by looking at the frequency of the lowest harmonic shown on the spectrum. Disregard any apparent "harmonics" at 0Hz–these are inaudible.)

9. In this scenario, *vibrato* refers to a sinusoidal variation in the breath pressure. Flutists produce this effect by varying their breath pressure at a rate on the order of 6Hz. Investigate the effect of changing the "Vibrato Gain" and "Vibrato Frequency" sliders. Which ranges of vibrato gains and frequencies are physically reasonable, and which ranges would not normally be played by flutists? At what maximum rate can you vary your own breathing?

10. Adjust the sliders and number boxes so that they match the `pd` patch shown in Figure 5. Now you will learn about *subpatches*, which are `Pd` patches that are embedded within higher-level patches. Double-click on the subpatch `pd cc`. Here you will find additional sliders that control more unusual parameters of the model. Look at the spectrum as you adjust the "Noise Gain" slider back and forth between minimum (left) and maximum (right). If you look carefully, you should notice that the spectrum looks chaotically grainy in between harmonics when the noise gain is adjusted to its maximum value. You are actually observing the effects of the broadband noise. You should hear that the output sounds more like static or poor radio reception as you increase the noise gain.

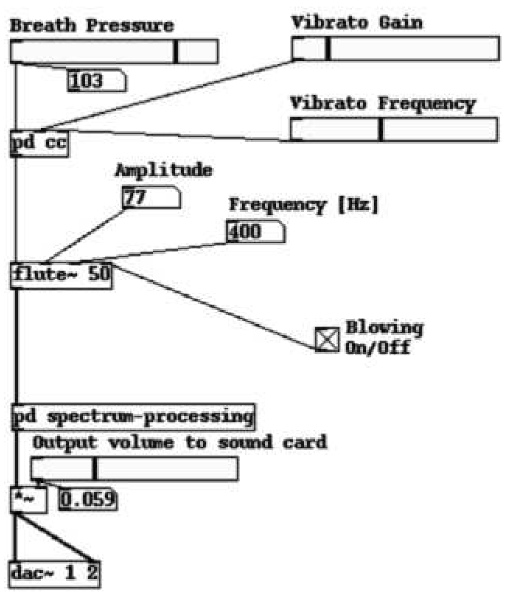

Figure 5: Sample settings for the main-level `pd` patch `flute.pd`

7 Problems

1. Many woodwinds, such as the clarinet, are better modeled with a closed tube end at the head of the instrument [3]. Use the expression

$$k = \frac{r_1^2 - r_2^2}{r_1^2 + r_2^2}. \tag{4}$$

   to calculate the reflectance k_{closed} for a closed tube end.

2. Do pressure waves have a node or an anti-node at a closed end?

3. Redraw the signal flow diagram shown in Figure 3 so that it models a woodwind instrument with a closed end at the head.

4. How does this change affect the fundamental frequency of the instrument?

References

[1] P. R. Cook, “A meta-wind-instrument physical model, and a meta-controller for real time performance control,” in *Proceedings of the 1992 International Computer Music Conference, San Jose*, pp. 273–276, Computer Music Association, 1992.

[2] P. de la Cuadra and B. Fabre, “Analysis of jet instability in flute-like instruments by means of image processing: effect of the excitation amplitude,” in *Proceedings of the International Symposium on Musical Acoustics (ISMA-04), Nara, Japan*, 2004.

[3] M. E. McIntyre, R. T. Schumacher, and J. Woodhouse, “On the oscillations of musical instruments,” *Journal of the Acoustical Society of America*, vol. 74, pp. 1325–1345, Nov. 1983.

[4] J. O. Smith, *Physical Audio Signal Processing*, `http://ccrma.stanford.edu/~jos/pasp/`, Aug. 2007, online book.