# A REAL-TIME DSP IMPLEMENTATION OF A FLUTE MODEL

*Vesa Välimäki, Matti Karjalainen, Zoltàn Jànosy* and *Unto K. Laine*

Helsinki University of Technology, Acoustics Laboratory, Otakaari 5 A, SF-02150 Espoo, Finland
and CARTES (Computer Arts Centre at Espoo), Ahertajankuja 4, SF-02100 Espoo, Finland

## ABSTRACT

This paper presents a new, computationally efficient semi-physical time-domain model of the flute. We have used the traditional method of modeling the one-dimensional wave propagation as a transmission-line, where all the losses as well as reflection and dispersion effects have been lumped to linear filters at the ends of the line. The model is excited by white noise, and the nonlinear interaction between the excitation and the tube resonator has been modeled by a static memoryless nonlinearity. The model is simple enough to be computationally efficient, but has retained many important features of a real instrument. The real-time implementation has been done on a Texas Instruments TMS320C30 floating-point signal processor using a sampling rate of 44.1 kHz.

## 1. INTRODUCTION

Physical modeling of acoustic musical instruments has recently become a field of growing interest in musical acoustics. A traditional technique for achieving natural sounding synthesis is the use of a digital transmission-line model of the wave propagation [e.g. 1]. It leads to a highly efficient algorithm, which is desirable since the implementation of a physical model should run in real-time to be useful in musical applications. The transmission-line approach suits well for modeling vibrating structures that contain wave propagation essentially in only one dimension. The strings of the guitar and tubes in wind instruments are typical examples where the vibration is of this type.

We are presenting a transmission-line model for the modern flute – also called the Boehm flute – and its real-time implementation on a digital signal processor. The work is based on previously reported real-time models of the vocal tract [2] and guitar string [3, 4] and on the knowledge of the mechanism of sound production by the flute. A good summary of the studies on the physics of the flute and a detailed bibliography can be found in a book by Fletcher and Rossing [5].

The modeling is done in the time-domain. One of the advantages of this approach is that the starting transients of the sound will automatically correspond to those of a real instrument. Also, any variations in the excitation will affect the resulting sound in a realistic fashion. In addition time-domain modeling allows us to easily handle the nonlinearity that is a most important feature of flutes and other wind instruments.

The bore of the flute is modeled by a digital transmission-line, which is a bidirectional delay-line with simple digital low-pass filters at both ends. The model is excited with white noise, and the nonlinear interaction between the excitation and the tube resonator is modeled by a static nonlinear element.

Many features typical to the sound of a real flute can be observed in the sound generated by our computational model. For example, the attack transient of the model brings to mind a real flute and it is possible to perform overblowing, i.e., transition to a higher mode, by changing certain parameters. The implementation of the flute model runs in real time on a single TMS320C30 floating-point signal processor using a sampling rate of 44.1 kHz.

## 2. DIGITAL TRANSMISSION-LINE MODEL OF THE TUBE

A digital transmission-line is used for modeling the vibrating air column inside the tube. The model does not take into account the effect of every single finger hole in the bore. That would turn the model into a much more complicated system, since there are 15 holes in the bore in addition to the embouchure hole and the end hole (the C hole), and each one of the open holes acts as a high-pass filter in relation to the pressure wave propagating inside the tube. They should be handled as short open side branches in the transmission-line. The closed holes should be included in the model as closed side branches. The complexity of such a model would prevent real-time implementation.

Instead we have assumed that the losses and dispersive characteristics of the tube are so minor that all such effects can be lumped, together with the reflection properties, to linear filters at the ends of the tube. The rest of the transmission-line can be implemented as a pure delay-line, which is simply a sequence of unit delays. The length of the delay-line corresponds to the effective length of the tube of a real flute.

Fig. 1 shows the principle of our flute model. Filters $F_F$ and $F_B$ bring about the damping and dispersion that is directed to the signal on the way from one end of the tube to the other and also the reflection properties of the corresponding end. Here we have used the frequency-domain point of view and have tried to simulate the frequency response of the flute with the frequency characteristics of the filters. The impulse response of the filters does not imitate in detail the reflection function of the tube ends, but it still is asymmetric and causal as the real reflection function is supposed to be [6].

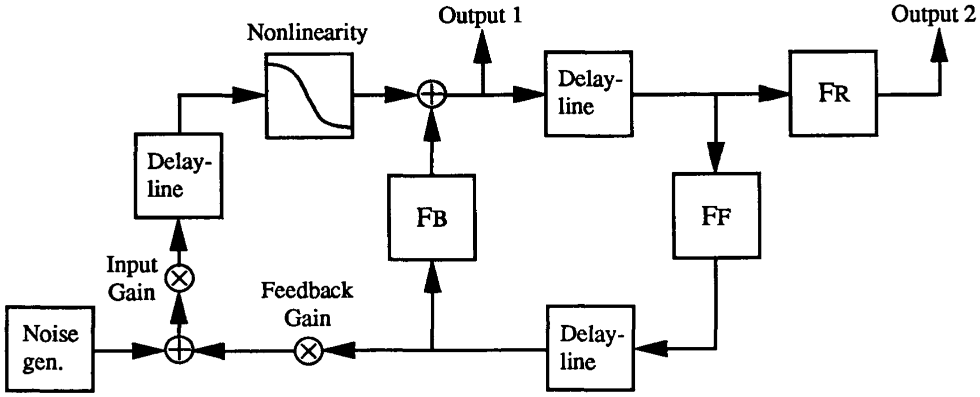


*Fig. 1. The block diagram of the flute model. $F_F$ and $F_B$ are low-pass filters that represent the reflection characteristics of the tube ends, dispersion and losses due to the walls of the bore. The filter $F_R$ models the radiation properties of the first open finger hole. The output of the model is computed as a weighted sum of outputs 1 and 2.*

## 3. APPROXIMATION OF VARIABLE LENGTH TUBE BY AN FIR INTERPOLATOR

Delay-lines consisting of an integer number of unit delays are not adequate for representing lengths required for playing the scale of equal temperament, because not all the needed lengths can be presented if a fixed finite sampling frequency is used. For that reason a way to vary the length of the tube continuously must be found.

Our solution to the continuous length variation problem with a fixed sampling frequency is to design an FIR filter that produces the fractional delay [2, 3, 4]. First the signal is modeled by a continuous function, a polynomial, which matches with the known sample values, and then, by using this polynomial, an approximation of the signal value between two samples can be computed. This method is known as Lagrange interpolation. Using this type of interpolator it is possible to change the length of the tube while playing without disturbing audible effects.

The Lagrange interpolator can be interpreted as an N-tap FIR filter. When the number of taps is $N = 2k$ ($k = 1, 2, ...$) the transfer function of the filter is defined by the formula:

$$L_{N,x}(z) = \sum_{i=-N/2+1}^{N/2} \lambda_i(x)\, z^{-i} \tag{1}$$

where $0 \leq x < 1$ is the fractional delay. The filter coefficients $\lambda_i(x)$ are given in the form:

$$\lambda_i(x) = \prod_{j=-N/2+1,\, j\neq i}^{N/2} \frac{x-j}{i-j} \tag{2}$$

The interpolator used in this implementation is a fourth order filter, the amplitude response of which is flat at low frequencies but falls off at the high end. The attenuation at high frequencies depends on the argument $x$. In the worst case, when $x = 0.5$, there is a zero at the folding frequency. According to an analysis of the fourth order Lagrange interpolator given in [2], the maximum error in the amplitude response below 25% of the sampling frequency is less than 1 dB while $0 \leq x < 1$. The sampling frequency used in the model is 44.1 kHz, which means that the accuracy of the interpolator is adequate for frequencies up to about 10 kHz. To achieve better accuracy and larger bandwidth, an FIR interpolator of higher order should be chosen.

Also the phase of the interpolator differs from an ideal fractional delay element. The largest errors appear close to the folding frequency with argument values $x = 0.4$ and $x = 0.6$, and there is no phase error when $x = 0.5$.

The fractional delay can also be approximated by IIR filters [1]. All-pass filters are considered a good solution for fixed fractional delay, but for dynamically varying tube lengths we prefer the FIR interpolator.

## 4. THE PRINCIPLE OF SOUND PRODUCTION IN THE FLUTE AND IN THE MODEL

### 4.1. Nonlinear Coupling of the Tube and the Air Jet

The flute belongs to a group of instruments called self-excited oscillators [5], which denotes that the excitation does not have to contain any kind of variation with time. The oscillation is produced by a nonlinear interaction of the air jet and the tube resonator. The disturbance at the embouchure hole due to the vibration inside the bore causes a transversal sinusoidal wave motion in the air jet, and this is why the jet is alternately directed away from or into the hole. When the distance of the lips of the player and the speed of the jet are properly adjusted, the oscillation arises and will be maintained.

The interaction of the tube and the air jet can be described in terms of a nonlinear function called the sigmoid function. It determines the pressure wave going into the tube as a function of the volume flow at the embouchure hole. The shape of the function can be derived from the profile of the air jet [5] and it can be presented in the form [6]:

$$F(f) = h + k \tanh(f) \tag{3}$$

where $f$ is the value of the volume flow advancing towards the embouchure hole, $k$ is a gain term, and $h$ describes the deviation of the center line of the air jet from the wedge. By changing the value of $h$ it is possible to produce asymmetrical waveforms or – in other words – sounds that also have even harmonics in their spectrum. When $h$ is zero, the spectrum of the output of the model has very weak even harmonics.

Since the air jet has to travel a short distance from the player's lips to the edge of the embouchure hole, there has to be an additional time delay in connection with the nonlinearity. The speed of the disturbance in the jet is actually slower than that of the jet itself [6]. Therefore, the delay-line before the sigmoid function is longer than it would be if it was derived from the physical distance.

By changing the length of the delay-line it is possible to perform so called overblowing, that is, a transition to a higher mode of oscillation. This is achieved by shortening the length of the line by an integer ratio [6]. With smaller changes to the time delay, some slight changes to the pitch take place.

### 4.2. Mode Locking

The resonance frequencies of a tube resonator are almost, but not exactly, harmonically related, because the effective length of the vibrating air column is frequency-dependent. However, the steady state tones of the flute have a repeating waveform – when no vibrato is used – indicating strict harmonicity. The frequency and phase of every mode of the tube are locked to those of others in such a way that the frequencies are forced to have an integer ratios. This phenomenon, called mode locking, is caused by the nonlinear coupling of the excitation and the resonator [5].

Due to the nonlinear phase function of the first-order IIR filters in our model, the overall length of the transmission-line is dependent on the frequency. At this point there is an analogy between the digital transmission line and a real tube. In our model the mode locking is achieved by the memoryless static nonlinear function. The output of a function like this has the same period length as the input and therefore always has a harmonic spectrum, but the effect that the nonlinear function has on the spectrum of its input depends on the amplitude. As can be concluded from Eq. 3 the sigmoid function behaves nearly like a linear gain element when the amplitude of the input is small, but the nonlinearity increases with amplitude. When the input amplitude is very large, the value of the function changes from maximum to minimum very rapidly and the resulting waveform is nearly a square pulse.

### 4.3. The Vibrato

An essential feature of the tones played by the flute is the vibrato. The player varies the blowing pressure in such a way that there is an approximately sinusoidal variation in the amplitude of the input. The amplitude of the vibrato is about 10% of the blowing pressure and the frequency around 5 Hz [5].

In the model there is a simple sine wave oscillator modulating the gain of the noise excitation. In the output the effect of this modulation is not that of the pure AM, since the harmonic content of the sound varies with the amplitude. This action is similar to that observed in the sound of a real flute and it is often called timbre vibrato. Of course, it is the smoothly clipping nature of the sigmoid function that causes this effect in our model.

### 4.4. Attack Transients and Decay

The model handles the starting transients and decays automatically without any serious problems, if the parameters of the model are properly chosen. The sound begins with setting the input gain of the model to a value large enough (see Fig. 1).

The simulation of the very beginning of the attack is problematic in our model because the memoryless nonlinearity with a delay before it does not describe the behaviour of the air jet precisely enough. Also, the idea of lumping the dispersive effects of the finger holes to the ends of the transmission-line prevents the signal from being smoothed during the first pass in a realistic manner. Still it is possible to produce a relatively natural attack.

When the input is turned off, the signal circulating in the two-way delay-line does not go through the nonlinearity any more, and the model turns into a linear and passive system. The signal uses the direct path from the lower delay-line to the upper one (see Fig. 1). After this the amplitude of the signal attenuates exponentially due to reflection filters, and so does the amplitude of the output. While the amplitude is changing the signal is also low-pass filtered. To conclude the effects to the output after the input is switched off, the amplitude of the output exponentially approaches zero and the output waveform approaches that of a sine wave in such a way that the higher harmonics are attenuating faster than the lower ones.

## 5. EVALUATION OF THE SOUND QUALITY OF THE MODEL

To sound natural, the tones the model produces must resemble those of a real instrument in certain ways. The attack transient must have similar characteristics as those of the real sound. It means that the spectrum level of each harmonic component must vary as a function of time in a realistic fashion. Also, the steady state spectrum must have proper characteristics, i.e. the levels of the harmonic components and the level of the noise must be natural.

We have measured the envelopes of the harmonics during the starting transient in the sound of a real flute. Also, we have analyzed the steady state spectrum of the flute and separated the harmonic components from it to be able to estimate the spectral shape of the noise component.

It seems that our approximation of the excitation as white noise is not bad. The background noise that is present in the real sound has somewhat flat spectrum. The steady-state sound of the model is easier to calibrate, than the attack transients, according to the real sound of a flute. The coefficient values of the filters $F_F$ and $F_B$ can be estimated from the decay (or impulse response) of the sound of a real instrument by means of LPC analysis.

It has not been possible for us to get the steady-state sound to be identical with that of the flute, because it is difficult to find a satisfying compromise between even harmonics with high amplitudes and a relatively steep slope of the overall spectrum. Still, the produced synthetic sound is remarkably flutelike and an average listener would not mistake it for any other wind instrument.

## 6. IMPLEMENTATION OF THE MODEL

For the real-time implementation we used the Texas Instruments TMS320C30 floating-point digital signal processor connected to an Apple Macintosh II host computer. The software system used for developing C30 assembly code and high-level control programs is the QuickC30 environment [7]. It is an object-oriented DSP programming environment based on Common Lisp and CLOS. The main idea in QuickC30 has been to integrate the C30 assembler to the Lisp language to achieve flexibility and interactivity together with high runtime efficiency.

The control of the parameters of the model is done by a special control program written entirely in CLOS. It performs a transformation from standard musical notation to control parameters and timing information needed to play musical pieces with the model. The final control parameter array is produced in several phases, but the scheduler included in the flute model program reads the sequence in real time. It is also possible to control the model by MIDI, although adjusting a large number of special parameters at the same time is not well supported in MIDI.

The circular addressing mode of the TMS320C30 made it possible to easily implement the delay-lines in an efficient way, because it eliminates the need to explicitly move data. In filtering routines the parallel multiply/add instructions were exploited.

For maximum efficiency the sigmoid nonlinearity was realized by a look-up table, using linear interpolation between table values. Another way would have been to approximate the function by a polynomial. The use of a look-up table was chosen because the computational cost does not increase as a function of accuracy.

The excitation noise is produced by a pseudo-random number generator which uses a new, extremely fast algorithm for generating uncorrelated noise samples. The algorithm makes use of special properties of the TMS320C30 instruction set. The spectrum of the noise it produces is flat, and also the amplitude distribution of the noise samples is flat enough for this purpose. The algorithm has not yet been thoroughly analyzed and the details will be published later. For the use in the flute model also any other relatively fast random number generator with proper characteristics will do.

One of the problems we have faced in the model is the growth of the DC component in the transmission line. The increasing DC level is caused by the fact that with some parameter values the DC gain of the loop in our model is greater than one, and although the pseudo-random sequence used as input does not contain a DC offset in the long run, it is possible that during a short period the time average of the sequence is nonzero. The DC component had to be removed because once it rises it will never disappear but causes severe numerical problems in the algorithm. Our solution to this problem is a first order IIR filter, the magnitude response of which is zero at 0 Hz and flat at frequencies over about 50 Hz. This filter is located after the nonlinear element in our model.

The basic flute algorithm uses a little more than 50% of the computation power of the TMS320C30 processor when the sampling frequency of 44100 Hz is used.

## 7. CONCLUSION

The flute model presented in this paper is a semi-physical model of the real flute. It is simple enough to be computationally efficient, but has retained many essential aspects of the real musical instruments. The real-time implementation on a single DSP chip makes it possible to use the model for experimentation in computer music or psycho-acoustic tests. In the real-time model any parameter can be changed independent of the others while playing. This is likely to help us to better understand how different parameters affect the sound of the flute, since in the real instrument it is extremely difficult to change only one parameter at a time.

It is possible to further improve the model by adding a more detailed description of the finger holes. Especially the inclusion of the octave hole would make the high tones more natural. This can be done by dividing the transmission-line into two parts and placing a high-pass filter between the parts. In addition, the white noise used for excitation could be filtered to model the spectrum of the volume flow signal in the real flute.

Our model can also be used for simulations of other flutelike instruments, such as the panpipes, the recorder and different kinds of flue pipes. The required parameter values have to be estimated with the help of measurements of the sound of those instruments.

The subjective sound quality of the present model is very good. New, faster signal processors will make it possible to add more details to the model so that it will produce even more natural sound.

## ACKNOWLEDGEMENT

This work has been supported by the Academy of Finland.

## REFERENCES

[1] D. A. Jaffe, J. O. Smith, "Extensions of the Karplus-Strong Plucked-String Algorithm," *Computer Music Journal*, vol. 7, no. 2, pp. 56-69, 1983.

[2] U. K. Laine, "Digital Modelling of a Variable Length Acoustic Tube," in *Proc. Nordic Acoustical Meeting (NAM-88)*, Tampere, The Acoustical Society of Finland, pp. 165-168, 1988.

[3] M. Karjalainen, U. K. Laine, "A Model for Real-Time Sound Synthesis of Guitar on a Floating-Point Signal Processor," in *Proc. IEEE ICASSP-91*, Toronto, pp. 3653-3656, 1991.

[4] M. Karjalainen, U. K. Laine, T. Laakso, V. Välimäki, "Transmission-Line Modeling and Real-Time Synthesis of String and Wind Instruments," in *Proc. Int. Computer Music Conf. (ICMC-91)*, Montreal, pp. 293-296, 1991.

[5] N. H. Fletcher, T. D. Rossing, *The Physics of Musical Instruments*, New York, Springer Verlag, pp. 426-466, 1991.

[6] M. E. McIntyre, R. T. Schumacher, J. Woodhouse, "On The Oscillations of Musical Instruments," *J. Acoust. Soc. Am*, vol. 74, no. 5, pp. 1325-1345, 1983.

[7] M. Karjalainen, "Object-Oriented Programming of DSP Processors: A Case Study of QuickC30," in *Proc. IEEE ICASSP-92*, San Francisco, 1992.