Physical modelling of a stiff string
by numerical integration

Michael Kurz
Technische Universität Berlin, Institut für Kommunikationswissenschaft
kurz@gigant.kgw.tu-berlin.de

Bernhard Feiten
Deutsche Telekom, Technologiezentrum
feiten@fz.telekom.de

Abstract
An algorithm for real time sound synthesis by physical modelling is presented. Starting from the differential equation of a string with harping stiffness and damping, a simple rule for computing the state of a virtual string is derived. Several methods of excitation are explored which allow the production of complex sounds.

1 Introduction
The use of computer simulation of the mechanics of musical instruments for the generation of sound, so called physical modelling, results in timbres with a "natural" feeling. Also, the parameters used in this kind of synthesis are intuitive and therefore easy to manage as they have direct correspondences in the physical world [1]. With today's technology physical modelling synthesis of tones produced by a vibrating string can be carried out in real time as long as the algorithm is carefully designed to be computationally efficient. The most common method used today, waveguide synthesis [2, 3], uses delay lines and filters, and several synthesizers employing this principle are commercially produced.

The method of direct computation of the displacement state of a string in real time, presented here, has not been much explored so far. The model consists of a virtual string, the shape of which is determined for every step in time. Numerical integration of the differential equation which determines the behaviour is used to compute the state of the string based on its previous state. Because the displacement of every point on the string is known, excitation can be applied to and sound output taken from any point on the string without extra computation. For excitation a separate physical model, for example of a hammer, can be connected to the string and the force between the two can be determined relatively easily. This force will affect the movement of the string as well as the exciting system and very realistic sounding effects can be produced.

The aim of the present work was to develop a program which would allow interactive control of the sound synthesis process in real time so that the system could be used like a virtual musical instrument. To this end a graphical user interface as well as MIDI input was added to the sound generation routines. In this way even non-technical people can experiment with the range of timbres available by adjusting the values of the parameters which control the behaviour of the model. This virtual string
2 The continuous string

The model is based on the differential equation describing the forces acting on and in the string [4]. It is assumed to behave linearly, which is a good approximation for most musical instruments where displacement is very small relative to the length of the string. Only transverse waves in one plane are considered. The string has a mass per unit length \( \mu \) and is homogeneous. A tension \( T \) is applied to the string. The bending stiffness \( I \) can be non-zero, leading to dispersion in the travelling waves. This has the effect of raising the frequencies of the harmonics and results in a metallic timbre. Damping is provided for with three parameters \( c_0, c_1, c_2 \) which differently affect the vibrational modes present on the string and thereby determine how the spectrum changes over time. A driving force which is variable in time and place can be applied, for instance for excitation.

Thus the differential equation, with \( y(x,t) \) being the displacement, is:

\[
\mu \ddot{y} = Ty'' - EIy'' + c_0 \dot{y} + c_1 y'' + c_2 y'''' + F_{ext}
\]

(1)

Its solution is the superposition of harmonic travelling waves. Taking a string of length \( L \) with two fixed ends, the standing waves (vibrational modes) have wavelengths:

\[
\lambda_n = \frac{2L}{n}, \quad n = 1, 2, 3 \ldots
\]

(2)

Their frequencies are:

\[
\omega_n = \sqrt{\left( \frac{n^2}{L} \right)^2 \frac{T}{\mu} + \left( \frac{n^2}{L} \right)^4 \frac{EI}{\mu}}
\]

(3)

3 The discrete string

To simulate the continuous string using a digital computer it is necessary to sample its state in time and space. This is done by representing the string by a number of point masses which exert forces on each other and whose displacement is determined at time steps separated by the sampling period \( \Delta t \). This representation is an approximation to the continuous string and its solution will be an approximation to the continuous solution. The error will be greatest for those vibrational modes whose wavelength is close to twice the mass spacing \( \Delta x \) and whose frequency is close to half the sampling frequency, i.e., the highest harmonics in the signal.

A system of \( n \) point masses with one degree of freedom each can have no more than \( n \) vibrational modes. Therefore the number of string elements used in the simulation determines the number of harmonics in the resulting audio signal.

To discretise the differential equation the simple Forward-Time-Centre-Space rule is used. Writing \( y^{(i)}_j = y(x = j\Delta x, t = i\Delta t) \):

\[
\frac{\partial y}{\partial t} \bigg|_{j,n} = \frac{y^{(i+1)}_j - y^{(i)}_j}{\Delta t}
\]

(4)

This leads to an explicit expression for calculating the new string state, i.e. using just known values:

\[
y^{(i+1)}_j = b_0 y^{(i)}_j + b_1 (y^{(i)}_{j-1} + y^{(i)}_{j+1})
+ b_2 (y^{(i)}_{j-2} + y^{(i)}_{j+2})
+ b_3 y^{(i-1)}_{j-1}
+ b_4 (y^{(i)}_{j-1} + y^{(i)}_{j+1})
+ b_5 (y^{(i)}_{j-2} + y^{(i)}_{j+2})
+ b_6 F^{(i)}_j
\]

(5)
The coefficients $b_0 \ldots b_n$ are simple functions of the string's physical parameters, the sample period and the mass spacing. They are constant as long as the string parameters are not changed, but this can be done while the synthesis is in progress to give real-time control.

$$n-1$$

$$n$$

$$n-1$$

$$\frac{t/\Delta t}{x/\Delta x} = j-2 \quad j-1 \quad j \quad j+1 \quad j+2$$

Figure 2: Dependence of a new value on known old values

3.1 Computational cost

As can be seen from equation (5), the required computing power (without driving force) is 9 addition and 6 multiplication operations or 16 multiply-accumulate operations (MAC) per sample period for each string element. So, on a signal processor, the required processing power for a string of $t$ elements is $P = 101.7 t \frac{[\text{ops/sec}]}{}$. The additional processing time required for excitation, overloads and input/output is small by comparison. On a Silicon Graphics Indy workstation a string of 35 elements can be computed in real time at a sample rate of 32 kHz.

3.2 Deviation of harmonic frequencies

Solving the difference equation (5) leads to an expression for the frequencies of the resonant modes of the discrete string. Comparing these to those in the continuous case shows that in the simulation, the frequencies of the harmonics are always somewhat too low. This is noticeable in the sound output but does not detract much from the perceived quality. The lowering of the frequencies of the harmonics can be partly compensated by adding a small amount of bending stiffness which has the effect of raising their frequencies. However, the highest partials will always be too low in frequency. The errors in the simulation become smaller as more processing power is used by increasing the sample rate and the number of string elements.

![Graph showing frequencies of harmonics in discrete strings](image)

Figure 3: Frequencies of harmonics in discrete strings

3.3 Stability

The approximations necessary for simulation also lead to problems of numeric stability. By definition the system is only stable as long as the damping is negative. This is also true for the continuous string but negative damping is impossible due to the laws of physics. In the simulation, however, the system will also be testable for some other choices of parameters, particularly when the highest partial (wavelength $2\Delta x$) has a frequency higher than half the sample rate. This will occur when the fundamental frequency is chosen too high with regard to the sample rate and number of elements. Within this limit the fundamental...
frequency can be set to any value by adjusting the tension and/or the stiffness.

4 Excitation

To set the string in motion a force is applied at one point along its length. In the simplest case the force acts only for the duration of one sample, i.e. it is an impulse. More interesting is the use of another physical model which interacts with the string for a length of time while they are in contact, such as a virtual hammer. The position at which the force is applied to the string as well as the parameters of the hammer will affect how much the various resonant modes of the string are excited and thus change the resulting timbre. The position of the point on the string whose movement is taken as the sound output also has a large influence on the signal's spectrum.

Figure 4: A virtual hammer

Applying a constant force representing gravity to the hammer results in it returning to make contact with the string after it has been thrown clear, again and again. This sounds very much like a ball bouncing on the string.

Similarly, a virtual spectrum as well as a simple bow have been constructed and used to obtain complex timbres.

5 Implementation

The algorithm has been implemented as a standalone program on a Silicon Graphics workstations. It comprises a full graphical user interface which allows real-time control of all parameters by means of faders, as well as MIDI input for control of pitch and excitation. Furthermore, it is possible to store and retrieve parameter settings and record the produced sound to disk. The executable and sample parameter files can be obtained on ftp://ftp.kw.tu-berlin.de/pub/krstring/

6 Conclusion

A new approach to physical modelling suitable for real-time synthesis was presented. Physically modeled excitation allows production of complex timbres with very "natural" sound. Metallic timbres can be obtained by simulating a string with bending stiffness or even a stiff bar without tension. Real-time control permits the program to be used as a virtual musical instrument.

References


