REAL-TIME EMULATION OF THE CLAVINET

Leonardo Gabrielli
Università Politecnica delle Marche,
Ancona, Italy

Vesa Välimäki
Dept. Signal Processing and Acoustics
Aalto University, Espoo Finland

Stefan Bilbao
School of Physics and Astronomy
University of Edinburgh, UK

ABSTRACT

A physical model has been developed for real-time sound synthesis of the Clavinet, an electromechanical keyboard instrument from the 20th century. The Clavinet has a peculiar excitation mechanism, relying on a tangent striking the string. The modeling paradigm chosen is waveguide synthesis and this paper suggests several novel techniques, such as a polynomial excitation pulse model and a beating generator, both of which have parameters depending on key velocity. Realistic emulation of the release part of Clavinet tones is based on a decrease in the decay rate of the tone and lengthening of a delay line, which corresponds to the physical string. A real-time implementation on Pure Data demonstrates the efficiency of proposed model.

1. INTRODUCTION

The analysis and synthesis of musical instruments by means of physical models allows to construct a wider knowledge on acoustics and their physical behavior, to bring them to a new digital life and make it easier to reproduce their sound with high detail and to deploy them to a vast number of musicians. Our work deals with the analysis of the Hohner Clavinet timbre and its reproduction by means of digital waveguide physical models. The name "Clavinet" refers to a family of instruments produced by Hohner between the 1960s and the 1980s; the most well-known model is the Clavinet D6 [1] . The digital waveguides prove computationally efficient while adequate to reproduce the tone of the real instrument. Up to the authors' knowledge there is no previous work done on the topic except from a first exploration of the FDTD modeling for the Clavinet string in [3]. The clavichord, an ancient stringed instrument which show similarities to the Clavinet, has been studied in [9]. Commercial software employing physical models for the Clavinet exist [1], but no specific knowledge on their algorithms is available.

[1]Our analysis is based on the Clavinet D6 in mint condition. We will generally refer to this specific model in our work as the Clavinet, neglecting the small differences that might exist between this and other models.

The paper is organized as follows. Section 2 deals with analysis of the Clavinet tone. Section 3 describes a physical model for the reproduction of its sound, while Section 4 discusses the Real-Time implementation of the model, showing its low computational cost. Section 5 concludes this paper.

2. INSTRUMENT ANALYSIS

The Clavinet is an electromechanical instrument with 60 keys and one string per key. The first 23 strings are wound and the remaining ones unwound, so that there is a small discontinuity in timbre between the twenty-third and the twenty-fourth notes. The excitation mechanism is based on a class 2 lever, where the force is applied through a rubber tip, called the tangent. The rubber tip strikes the string and traps it against a metal stud for the duration of the note, splitting the string into a speaking and a non-speaking part.

The vibration of the speaking part of the string is captured by two magnetic transducers, similar to guitar pickups, while the non-speaking part of the string is highly damped by a yarn winding. When the key is released the whole string speaks, hence making the pitch lower, but briefly, as the yarn damping is then applied to the whole speaking length of the string. There are two pickups: one lies close to the far string termination (the bridge pickup), and another at a varying distance closer to the string center (the central pickup).

The Clavinet also includes an amplifier stage, with tone control and pickup switches. This paper does not cover the modelling of these components.

2.1. Excitation mechanism

A physical model of the Clavinet, including the excitation mechanism is covered in [3]. In the present paper we will deviate from a strict physical model for the excitation because the DWG approach does not allow for exact reproduction of physical behavior and also because the experimental determination of the excitation wave shape using acoustic transducers is difficult because of key noise.

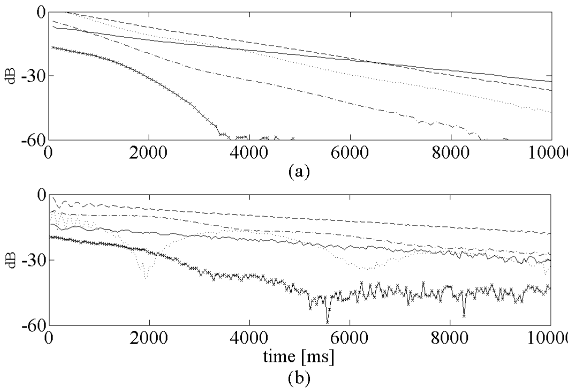

Figure 1. Harmonic decay for two different recordings of an $A\sharp 2$ Clavinet tone (a) *piano*, (b) *mezzoforte*. Harmonics are represented by lines: solid for 1^{st}, dashed for 2^{nd}, dotted for 3^{rd}, dash-dotted for 4^{th}, solid cross for 5^{th}.

For analysis and modelling of the string excitation mechanism, we used the signal provided by the pickups. This signal is, to a first approximation, a differentiated version of the displacement wave, thus approximating a velocity.

2.2. Tone analysis

Recorded tones serve as a database for the analysis of important features of Clavinet timbre and highlight some important aspects of its sound. A Clavinet tone is divided into the usual different phases: attack, sustain, release and decay. The attack is the transient phase caused by the tangent strike, while sustain corresponds to the free vibration of the speaking part of the string, kept trapped against the metal stud by the tangent. There is no specific damping mechanism for the string during these phases. When the key is released, there is a transient after which the string vibrates as a unit, making the pitch of the tone lower. Damping is now determined by the yarn winding at the leftmost termination, which, during the note decay, quickly damps the string, muting its sound.

There is a slight variation of the fundamental during the attack phase, but it is perceptually insignificant, being a shift of at maximum 1-2 cents. The over-all decay is two-stage, except for the highest notes which are very damped and therefore decay quite fast. Most of the harmonics show a linear decay on a dB scale, with generally very high T60, but some exhibit oscillating behavior as seen in Figure 1(b). There is no clear connection between pitch or key velocity and this phenomenon.

Inharmonicity has been measured for the Clavinet and compared to audibility thresholds extracted in [5]. From this comparison (see Figure 2) inharmonicity in the lowest tones clearly exceeds the audibility threshold and its confidence curve, making its emulation important in the string model.

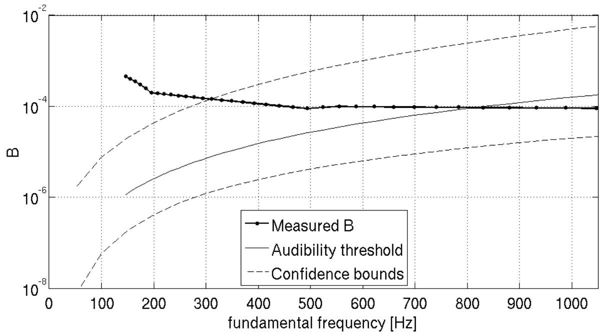

Figure 2. Measured inharmonicity coefficient for the Clavinet, against audibility thresholds from [5].

3. THE STRING MODEL

The strings are simulated using digital waveguide models. Although in principle, two delay line pairs should be used to simulate a string, in order to correctly model the two transverse string polarizations, the current model only uses one. This is motivated by efficiency concerns; the typical beating effect that is created by two coupled digital waveguides can be obtained by introducing a beating equalizer [7].

3.1. Model overview

The string model, depicted in Figure 4, is a waveguide model plus a beating equalizer and a pickup model cascaded outside the loop. The filters in the loop have the following transfer function:

$$S(z) = \frac{1}{1-(z^{-R}+r)(F(z)H_{loop}(z)H_d(z)z^{-(D_R-R)})} \quad (1)$$

where $F(z)$ is a first-order allpass fractional delay filter [4], $H_{loop}(z)$ is a one-pole filter, simulating frequency-dependent losses [8], and $H_d(z)$ is an allpass dispersion filter [4], designed using a novel method proposed in [2]. The tuning of the $H_{loop}(z)$ filter gain and pole coefficients has been done by ear tests. The gain has been set constant to 0.996 for all the keys, while the pole coefficient has been determined for several keys and is linearly interpolated over the whole keyboard range, showing a monotonic increase from -0.8 to -0.1. The delay line, denoted as z^{-L}, is split into $z^{-(D_R-R)}$ and rz^{-R}, to implement the ripple filter, introduced in [10]. This allows for frequency-dependent losses to have a more lively behavior as in real Clavinet tones. There seems to be no clear connection between tone pitch or dynamic and the ripple filter parameters, that are thus selected randomly in a range of typical values: the range from -0.005 to -0.001 for r and around $R = 0.31L$ for R, where L is the total delay-line length.

The output of the string model is filtered by a cascade of M beating equalizers in order to model the oscillating envelope of M different harmonics; the design of each follows that suggested in [7]. The gains in the equalizers vary according to a $g\,|\cos(2\pi f t)|$ function, with frequency and gain chosen according to a rule which is dependent on note number, again due to the fact that there seems to be no evident connection between key velocity and the occurrence of these beatings. Analysis of recorded tones suggests that the beating occurs only in harmonics higher than the second, with amplitudes ranging between 0 - 30dB and frequency ranging between 0.1 - 2Hz. The beating disappears gradually over the range from G3 to G4. A factor depending on the note number is applied to the amplitude gain accordingly. Our informal listening tests suggest that increasing M over a certain threshold (e.g. 3) results in no audible difference.

3.2. Excitation model

The excitation signal is fed to the string model described so far only once to produce sound. In our model, the excitation signal is produced by joining together a curved ramp and its reverse in order to obtain a pulse similar to the ones seen in many samples in the low to mid range. The curve is obtained by fitting a polynomial such as:

$$f(x) = a_P x^P + a_{P-1} x^{P-1} + \ldots + a_1 x + a_0 \tag{2}$$

(with P being the order of the polynomial) to some pulses extracted from the recordings. This signal is scaled by a gain and stretched by interpolation according to the player dynamic, making it shorter or longer. The length of the pulse is calculated according to the following physical assumptions: if the key velocity v is supposed constant during the tangent fall, knowing the initial distance d between tangent and stud, it is possible to calculate the required time in samples, N, for the pulse rise time according to:

$$N = \frac{f_s d}{v} \tag{3}$$

where f_s is the sampling frequency. This allows the model to be faithful to the physical behavior underlying the Clavinet tone. Key velocities are normally in the range 1 to 4 m/s, and are mapped to integers from 1 to 127, as per the MIDI standard. Figure 3 shows *piano* and *forte* excitation signals calculated with our method.

The polynomial coefficients were calculated from several least square error fits to some portions of signals extracted from the recordings. These signals have a smooth triangular shape, and represents the pickup output from the tangent hitting the string. Most of the recorded tones exhibit a similar pulse at the beginning of the tone, hence making this a good approximation for the string excitation produced by the tangent in most cases. Because the signal extracted from the pickups is the time derivative of the string displacement at the pickup position, when using its approximation as an excitation, it must be ensured that the wave variables in the digital waveguide are also time-differentiated approximations of the displacements of the Clavinet string. This allows differentiation to be avoided when emulating the effect of pickups.

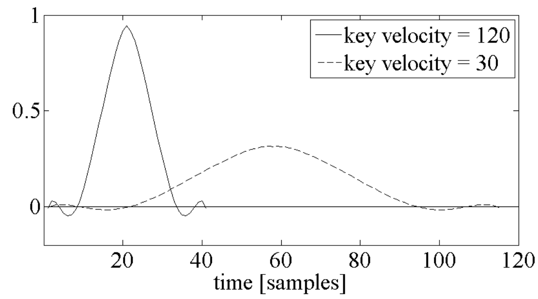

Figure 3. Excitation pulse signal for *piano* and *forte* tones.

3.3. Pickup model

A tone synthesized from the model described so far outputs a velocity wave from the end termination of the string. This is not physically meaningful. A basic pickup model is needed.

The velocity wave and the pickup output voltage are both time derivatives of the string displacement signal, except for additional filtering due to the nonideal nature of the pickup. A first approximation of the pickup transfer function is that of a low-pass filter with a cutoff frequency of approximately 5 kHz. Furthermore, the transducers pick up the signal of waves traveling in both directions. This can be implemented by taking as output the sum of two taps of the delay line, equally spaced around the central tap (which in turn inverts the wave). Equivalently, it can be shown that it is possible to send the delay line output to a feedforward comb filter with negative gain. We found this solution more practical for our implementations.

The comb filter delay K, is calculated as:

$$K = \frac{2 p_d D_t}{s_l}, \tag{4}$$

where D_t is the theoretical delay line length, p_d is the pickup distance from the right string termination and s_l is the string length.

The Clavinet allows to select the pickup signals separately, summed in phase or summed in counter-phase. This can be easily achieved by using two FIR comb filters in parallel, each corresponding to a different pickup position. Their outputs can be selected individually, summed, or subtracted.

3.4. Complete time-varying model

In order to model the different phases of a Clavinet note and other time-varying parameters, the previously-described model must be enhanced. During key release, and according to key release velocity, the $H_{loop}(z)$ filter coefficients must change. These change with a linear curve in order to reach the decay values. Decay values for the $H_{loop}(z)$ filter have been chosen by ear. Finally, to emulate the pitch

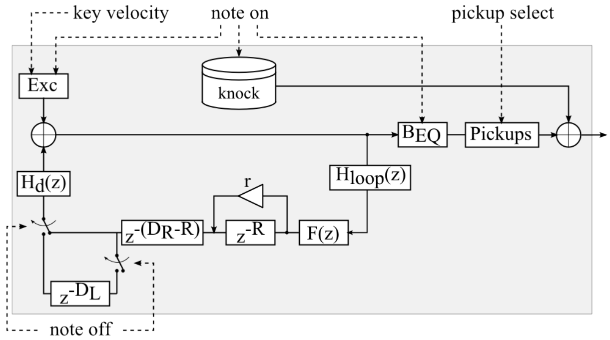

Figure 4. The proposed model.

change after key release, the delay line length must be increased instantly. This is done by adding an additional delay line at key release, of length D_L, calculated in order to obtain a -3 semitones change in the pitch, as it happens with the real instrument. A last improvement is given by triggering a soundbox knocking sound during note attack [9] and by adding it to the final mix through a velocity-dependent gain. The complete model is shown in Figure 4 .

4. REAL-TIME IMPLEMENTATION IN PURE DATA

Computer simulations, carried out in the Matlab prototyping environment, provided good quality results and motivated for an implementation of the string model on the Pure Data (PD) real-time-oriented platform [6], which is aimed at live music performances, and allows easy control through a MIDI keyboard. Currently, our PD implementation can run several string models on a common Linux PC platform with no perceivable latency. Our tests demonstrate that a polyphonic patch, running 4 instances of the string model requires an average 13% CPU load on an Intel Core2 Duo 1.6GHz laptop, allowing, theoretically for a 30-voice polyphony. Furthermore, the current PD implementation of the model only relies on the PD-extended package externals: this means that, in the future, if using custom-written C++ code to implement parts of the algorithm (e.g. the whole feedback loop), overhead for the computational cost can be highly reduced. This would leave headroom for additional complexity in the model.

5. CONCLUSION

This paper deals with the development of a DWG model for the Hohner Clavinet keyboard instrument. This instrument presents several new challenges and important differences with other instruments previously modelled, the first being the excitation mechanism and its faithful reproduction without having a direct measure of the tangent-string interaction. This has been overcome by dealing with its time-derivative and by making assumptions on the excitation signal by means of analyses of recorded tones. The model has been developed in Matlab, but a real-time prototype for the instrument has been created for the PD environment, with excellent results in both timbre quality and computational cost. Sound examples will be provided for further reference online at www.acoustics.hut.fi/go/icmc11-clavinet.

6. ACKNOWLEDGMENT

This research has been partially funded by the Academy of Finland (project no. 122815).

7. REFERENCES

[1] "Pianoteq cl1," 2010. [Online]. Available: http://www.pianoteq.com/commercial_addons

[2] J. S. Abel, V. Välimäki, and J. O. Smith, "Robust, efficient design of allpass filters for dispersive string sound synthesis," *IEEE Sig. Proc. Letters*, vol. 17, no. 4, pp. 406 – 409, April 2010.

[3] S. Bilbao and M. Rath, "Time domain emulation of the clavinet," in *AES 128th Convention, London, UK*, May 22–25, 2010.

[4] D. A. Jaffe and J. O. Smith, "Extensions of the Karplus-Strong plucked-string algorithm," *Computer Music Journal*, vol. 7, no. 2, pp. 56 – 69, 1983.

[5] H. Järveläinen, V. Välimäki, and M. Karjalainen, "Audibility of the timbral effects of inharmonicity in stringed instrument tones," *Acoustics Research Letters Online, ASA*, vol. 2, no. 3, pp. 79 – 84, April 2001.

[6] M. Puckette, "Pure data: another integrated computer music environment," in *in Proc. Int. Computer Music Conf.*, 1996, pp. 37–41.

[7] J. Rauhala, "The beating equalizer and its application to the synthesis and modification of piano tones," in *Proc. DAFx-07, Bordeaux, France*, 2007, pp. 181–187.

[8] V. Välimäki, J. Huopaniemi, M. Karjalainen, and Z. Jánosy, "Physical modeling of plucked string instruments with application to real-time sound synthesis," *J. Audio Eng. Soc.*, vol. 44, no. 5, pp. 331–353, May 1996.

[9] V. Välimäki, M. Laurson, and C. Erkut, "Commuted waveguide synthesis of the clavichord," *Computer Music Journal*, vol. 27, no. 1, pp. 71 – 82, Spring 2003.

[10] V. Välimäki, H. Penttinen, J. Knif, M. Laurson, and C. Erkut, "Sound synthesis of the harpsichord using a computationally efficient physical model," *EURASIP J. Applied Signal Processing*, vol. 7, pp. 934–948, 2004.