Commuted Piano Synthesis

Julius O. Smith III and Scott A. Van Duyne
Center for Computer Research in Music and Acoustics (CCRMA)
Department of Music, Stanford University, Stanford, CA 94305
jos@ccrma.stanford.edu, savd@ccrma.stanford.edu
Published in the Proceedings of the ICMC-95, pp. 319–326

ABSTRACT: The "commuted piano synthesis" algorithm is described, based on a simplified acoustic model of the piano. The model includes multiple coupled strings, a nonlinear hammer, and an arbitrarily large soundboard and enclosure. Simplifications are employed which greatly reduce computational complexity. Most of the simplifications are made possible by the commutativity of linear, time-invariant systems. Special care is given to the felt-covered hammer which is highly nonlinear and therefore does not commute with other components. In its present form, a complete, two-key piano can be synthesized in real time on a single 25MHz Motorola DSP56001 signal processing chip.

1 Introduction

In [11, 6], techniques were described for simulating plucked, struck, and bowed string instruments in which the resonating body of the instrument is *commuted* with the string in order to eliminate the high order digital filter which is otherwise necessary to simulate body resonances. In this technique, the string excitation (e.g., the "pluck" or "strike" force over time) is *pre-convolved* with the body impulse response, and the resulting waveform can be stored in a wavetable. To synthesize a plucked or struck string tone, the wavetable is simply "played into" the string. The resulting sound is that of a plucked or struck string with all the body resonances of the natural instrument, since they are pre-installed in the excitation. The sound quality is excellent for linearly plucked and struck strings. For bowed strings, the quality is quite good for smooth bowing styles that do not involve too much bow force. This paper describes an extension of this technique to the piano using a linearized model of the piano hammer which depends on striking velocity.

2 The Piano

The piano is an example of a *nonlinearly* struck string. It is simple in some ways and highly complicated in others. It is simple in that only the hammer velocity matters as a control variable when the string is struck—the finger that presses the key has no significant mechanical connection to the hammer after it is launched into flight toward the string. That means MIDI, for example, provides a sufficient representation for piano performance, and the dimensionality of control (aside from pedals) is confined to one degree of freedom per key, per time instant—the so-called "velocity" parameter.

2.1 String

Piano strings are fairly simple because they are uniform, tightly stretched, and nearly rigidly terminated. As a result, they are highly linear under normal playing conditions. The digital waveguide approach to string modeling [10] therefore works very well for the individual piano strings. The non-negligible stiffness of piano strings poses an increase in the expense of the implementation, resulting in the need for an allpass filter in the "string loop". Allpass filters of order $4-6$ or more are required for good results [7, 14]. Another complicating factor is the non-negligible coupling between strings that are hit by the same hammer [17]. There is also significant coupling among all the strings when the sustain pedal is down. To fully simulate the linear behavior of each string, it is necessary to couple [11] at least *three* digital waveguides together corresponding to the main types of wave propagation in and along the string (two transverse and one longitudinal). In key ranges in which the hammer strikes three strings simultaneously, *nine* coupled waveguides are required per key for a complete simulation. This paper will address the case in which only the *vertical* plane of vibration is simulated for each string, and only one string is implemented per key. In a reasonably high quality implementation, at least the correct number of strings should be implemented, since they are detuned and cause important beating and aftersound effects [17]. However, since extending the present discussion to multiple strings is straightforward, only the single-string case will be treated here.

2.2 Resonator

The soundboard and enclosure as a whole are simple in that they are largely linear, time-invariant components, but they are complex in that they are large. Large vibrating objects generally have many more resonant modes in the range of human hearing than do small objects. Also, waveguide propagation in the soundboard and enclosure is not confined to one dimension as it is in a string. That means a complete digital waveguide model of the piano would require two- and/or three-dimensional waveguide meshes [13] to model the resonating soundboard and piano enclosure. In sum, the sheer size of the piano and its soundboard lead to very expensive direct modeling techniques, even after accounting for the fundamental efficiency advantages of the digital waveguide approach. However, the commuted synthesis technique described below bypasses this difficulty and allows simple "sampling" of the soundboard/enclosure impulse response into a read-only memory which is "played" into the string in a manner modified by the hammer-string collision. The same technique applies equally well to the huge bank of sympathetically vibrating strings obtained when the sustain pedal is down [16].

2.3 Hammer

A more seriously complicating factor is the piano hammer. While only its velocity is necessary to specify its state completely prior to hitting the string, the string collision is highly nonlinear [12]. The nonlinearity comes from the felt covering the hammer: As it compresses, it acts like a spring whose spring-constant is rapidly increasing. Also, the hammer-string interactions are a function of both string and hammer motion, giving potentially complex cases such as the string hitting the hammer a second time after it has already fallen away from the initial strike. Apart from the hammer, the entire instrument can be very well approximated by a linear model. The main difficulty with nonlinearity in this context is that it prevents use of the commuted synthesis technique at first sight. This is because commutativity of system elements is only possible in general for linear, time-invariant elements. In previous work, we could relax the time-invariance requirement to some extent to allow for string vibrato. However, relaxing linearity is much more problematic, especially when the nonlinearity is this severe.

3 Commuted Piano Synthesis

It turns out commuted piano synthesis is possible with both high fidelity and low computational cost in spite of the nonlinear behavior of the hammer-string interaction.

The key observation is to note that the interaction between the hammer and string consists essentially of a *few discrete events* per hammer strike when the string is initially at rest. That is, the hammer-string interaction can be approximated as one or a few discrete impulses which are *filtered* using filters which depend on the hammer-string collision velocity. Figure 1 illustrates the qualitative behavior of the striking force [2, 12]. In this example, the three peaks in the force curve indicate that the hammer stays in contact with the string long enough for return pulse waves from the agraffe to compress the felt two more times before the hammer falls away the string. Also indicated in the figure are where three impulses may be located in order to synthesize the waveform as a superposition of filter impulse responses.

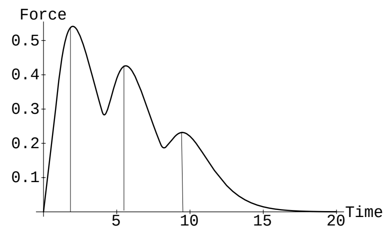

Figure 1: Example of overlapping hammer-string interaction force pulses. The vertical lines indicate the locations and amplitudes of three single-sample impulses which may be passed through small digital filters to produce the overlapping pulses shown.

To a large extent, the number of interaction impulses is determined by which string is being struck. Thus, given key number and hammer velocity, one can predict the amplitude and timing of all interaction impulses, for a string initially at rest. There is a slight unpredictability which we neglect having to do with the fact that when the hammer strikes an already vibrating string, the entire history of string vibration influences the exact details of the hammer-string interaction; however, this is a second-order effect which may not even be desirable. We still retain the superposition of the new strike response with any existing vibration, thus preserving the naturally varied colorations of successive strikes on a single string as is characteristic of a physical modeling technique. If unpredictability of the force pulses on restriking is deemed important, one may use random perturbations of the interaction impulse levels as a function of amplitude of vibration prior to the hammer strike. For greatest precision, of course, a rigorous piano hammer model may be run in parallel to compute the hammer-string interaction force in real time as a function of their relative velocities [1, 2, 12, 15].

The creation of a force pulse from a single impulse for a specific dynamic level is shown in Fig. 2. Without loss of generality, we consider force units; dividing by the string wave impedance gives the corresponding velocity injection for the string loop. Other physical variables may be chosen, as discussed in [10].

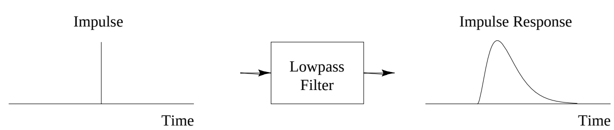

Figure 2: Creation of a single hammer-string interaction force pulse as the impulse response of a lowpass filter. The input to the filter is a single nonzero sample (impulse), and the output is the desired hammer-string force pulse. When the amplitude of the input impulse increases, the output pulse increases in amplitude and decreases in width, which means the filter is nonlinear. However, on each specific impulse, the filter operates as a normal linear, time-invariant filter. In this way, the entire piano is "linearized" with respect to each fixed hammer velocity.

3.1 Illustrative Implementation

One method of creating multiple force pulses from multiple impulses at a specific dynamic level is shown in Fig. 3. The multiple interaction impulses become multiple overlapping impulse *responses* which feed the summer, and the summer output is fed to the string.

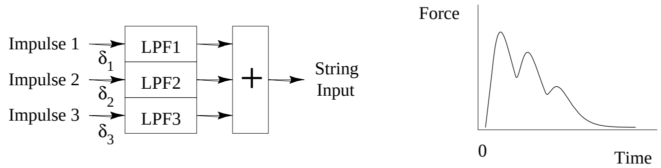

Figure 3: Creation of multiple hammer-string interaction force pulses as the superposition of impulse-responses of a bank of digital filters. The input to each filter is a single impulse, and the sum of their outputs gives the desired superposition of hammer-string force pulses. When the input impulses increase in amplitude, the output pulses become taller and thinner, showing less overlap.

At a specific dynamic level, we have obtained the critical feature that the model is linear and time invariant. That means we may now commute the soundboard/enclosure filter with not only the string, but with the hammer lowpass filter as well. These operations are carried out in going from Fig. 4, which shows a naturally ordered schematic diagram of the complete piano synthesis system, to Fig. 5, which shows the results of commuting the hammer-string assembly with the soundboard and enclosure.

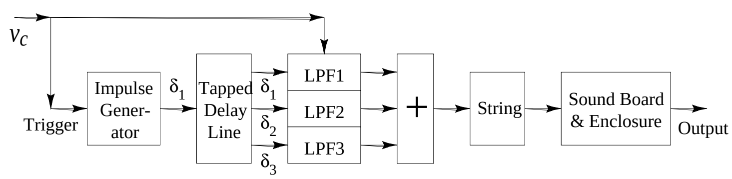

Figure 4: Piano synthesis as described using natural ordering of all elements.

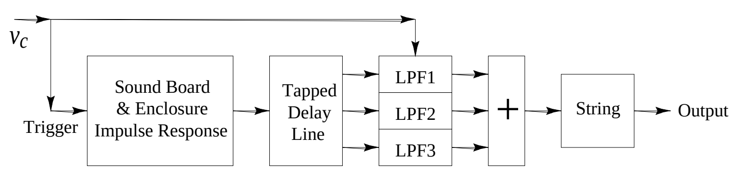

Figure 5: Piano synthesis using commuted ordering. The soundboard and piano enclosure are commuted such that we only need a stored recording of the impulse response of their series combination. The large digital filter required to implement the soundboard and the piano enclosure is thus removed. A change in the hammer-string collision velocity v_c changes the filters and triggers playback of the soundboard/enclosure impulse response.

While the commuted result is valid only for a fixed hammer-string collision velocity v_c, that is all we need. For different collision velocities, we simply alter the filters, denoted LPF1 through LPF3 in

Fig. 5, applied to the soundboard/enclosure impulse response. This works because the nonlinearity is confined to the hammer-string collision, and these are discrete, non-overlapping events which can be modeled individually using linear, time-invariant elements, indexed by collision velocity. Note, however, that if a "virtual piano key" is restruck before the excitation table has finished playing out, that playback must either be prematurely terminated (the low-cost solution), or multiple, overlapping playbacks must be supported, as in commuted bowed-string synthesis [11].

3.2 Excitation Factoring

It is typically more efficient to implement the highest Q resonances of the soundboard and piano enclosure using actual digital filters. By factoring these out, the impulse response is shortened and thus the required excitation table length is reduced. This provides a classical computation versus memory trade-off which can be optimized as needed in a given implementation. The explicit resonators can be conveniently implemented using parametric equalizer sections, one per high-Q resonance. In many practical situations, parametric eq sections may already be available in a separate effects unit.

A possible placement of the resonators is shown in Fig. 6. However, since all elements are linear and time invariant, they may be ordered arbitrarily. For example they could appear before the string. Having the resonators at the end, however, is convenient for defining *multiple outputs* having different spectral characteristics. Traditionally, resonators, equalization, dynamic comb filtering, and reverberation are implemented as post-processors, and these can all provide a diversity of outputs which can be panned individually into the stereophonic (or N-channel) sound-output stream for added sonic richness.

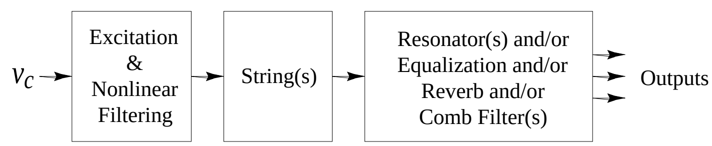

Figure 6: Example block diagram of a complete, commuted-piano synthesis system, including resonators which partially implement the response of the soundboard and enclosure, equalization sections for piano color variations, reverberation, comb-filter(s) for flanging, chorus, and simulated hammer-strike echoes on the string, and multiple outputs for enhanced multi-channel sound.

3.3 String Reverb

The sound of all strings ringing can be summed with the excitation to simulate the effect of many strings resonating with the played string when the sustain pedal is down. The string loop filters out the unwanted frequencies in this signal and selects only the overtones which would be excited by the played string. This is just another case of commuting the string with a resonator, where in this case, the resonator includes a bank of sympathetically vibrating strings. Sampling synthesis techniques can be used, for example, to synthesize the sound of all piano strings resonating at the same level.

4 String Interface

In a physical piano string, the hammer strikes the string between its two endpoints, some distance from the agraffe and far from the bridge. This corresponds to the diagram in Fig. 7, where the delay lines are drawn according to their physical interpretation.

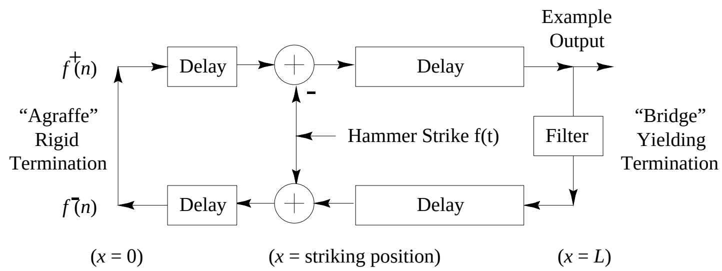

Figure 7: Illustration of string excitation in a filtered delay loop arranged to display the physical model interpretation. The delay lines contain samples of traveling force waves in this case, and other wave variables yield a similar diagram. The hammer-string interaction force pulse is summed into both the left- and right-going delay lines, corresponding to sending the same pulse toward both ends of the string from the hammer. One direction is negated relative to the other in a force wave simulation, while both are the same sign in a velocity wave simulation (but then the string terminations would be inverting).

By commutativity of linear, time-invariant elements, Fig. 7 can be immediately simplified to the form shown in Fig. 8, in which each delay line corresponds to the travel time in *both* directions on each string segment. From a structural point of view, we have a conventional filtered delay loop plus a second input (inverted) which sums into the loop somewhere inside the delay line. The output is shown coming from the middle of the larger delay line, which gives physically correct timing, but in practice, the output can be taken from anywhere in the feedback loop. It is probably preferable in practice to define the output as the delay-line *input*. That way, other response latencies in the overall system can be compensated to a maximum extent.

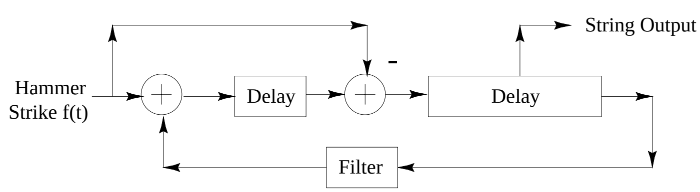

Figure 8: Diagram equivalent to Fig. 7, obtained by combining upper- and lower-rail delay lines.

An alternate structure equivalent to Fig. 8 is shown in Fig. 9, in which the second input injection is factored out into a separate comb-filtering of the input. The comb-filter delay equals the delay between the two inputs in Fig. 8, and the delay in the feedback loop equals the sum of both delays in Fig. 8. In this case, the string is modeled using a simple filtered delay loop, and the striking force signal is separately filtered by a comb filter corresponding to the striking point along the string. This factoring adds to the amount of memory needed, but (1) simplifies automatic loop calibration, and (2) the comb filter can be implemented elsewhere, such as in an effects unit. Post-processing comb filters are often used in reverberator design and in virtual pick-up simulation.

The comb-filtering can also be conveniently implemented using a second tap from the appropriate delay element in the filtered delay loop simulation of the string, as depicted in Fig. 10. The new tap output is simply summed (or differenced, depending on loop implementation) with the filtered delay loop output. Note that making the new tap a moving, interpolating tap (e.g., using linear interpolation), a *flanging* effect is available. Adding more moving taps and summing/differencing their outputs, with optional scale factors, provides an economical *chorus* or *leslie* effect. These

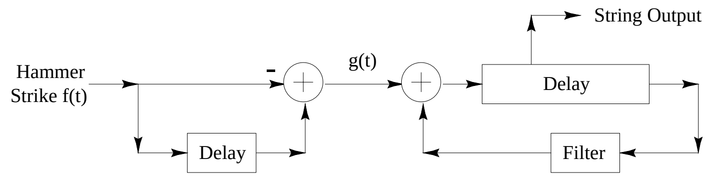

Figure 9: Diagram equivalent to Fig. 7, obtained by replacing the second string input by a separate comb-filter applied to a single input. It can be quickly derived by pushing the left-most delay in Fig. 7 through the summer on its left.

extra delay effects cost no extra memory since they utilize the memory that's already needed for the string simulation. While such effects are not traditionally applied to piano sounds, they are applied to electric piano sounds which can also be simulated using the same basic technique.

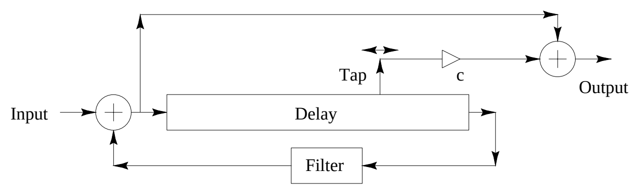

Figure 10: Use of a second delay-line tap to implement comb filtering.

It is also possible to eliminate explicit comb-filtering corresponding to the hammer striking position. The uniform spacing of the force pulses in the excitation signal $f(t)$ is the same as the delay needed for the striking-position comb filter. As a result, the physical force-injection signal $f(t)$ can be replaced by the comb-filtered version $g(t) = f(t) - f(t - \tau)$, where τ is the travel time from the striking point to the agraffe and back. The comb filtering can be applied to the excitation table prior to the shaping filter(s), or the shaping filter(s) can be designed to convert the excitation table directly into $g(t)$ rather than $f(t)$. In either case, the final excitation signal $g(t)$ simply drives a single filtered delay loop.

Perhaps it should be emphasized here that for medium to high quality piano synthesis, *multiple* filtered delay loops should be employed per key rather than the single-loop case discussed here. Each delay loop may correspond to a different string hit by the same hammer, a different polarization plane on a single string, or to a longitudinal wave. Thus, in a good piano synthesizer, there should be at least two filtered delay loops, tuned differently (unequal loop delays), both excited by $g(t)$ in some manner (e.g., they can each received an identical copy), and the delay loop outputs should be at least summed or else realistically coupled as described in [11].

5 Conclusions

A highly efficient computational model for the piano derived from an acoustic model was described. The hammer-string force interactions are modeled as discrete events which can be modeled as one or a few successive impulse responses of low-order digital filters. The soundboard and enclosure filtering is replaced by a look-up table using one or a few read-pointers per note. Further details are given in the companion paper [16], and related techniques are discussed in [3].

References

[1] X. Boutillon, “Model for Piano Hammers: Experimental Determination and Digital Simulation,” *J. Acoust. Soc. Amer.*, vol. 83, no. 2, pp. 746–754, Feb. 1988.

[2] A. Chaigne and A. Askenfelt, “Numerical Simulations of Piano Strings. I. A Physical Model for a Struck String Using Finite Difference Methods,” *J. Acoust. Soc. Amer.*, vol. 95, no. 2, pp. 1112-1118, Feb. 1994. “Numerical Simulations of Piano Strings. II. Comparisons with Measurements and Systematic Exploration of Some Hammer-String Parameters,” JASA vol. 95, no. 3, pp. 1631–1640, March 1994.

[3] D. A. Jaffe and J. O. Smith, “Performance Expression in Commuted Waveguide Synthesis of Bowed Strings,” *Proc. 1995 Int. Conf. Computer Music,* pp. 343–346, Banff.

[4] J. Laroche and J. L. Meillier, “A Simplified Source/Filter Model for Percussive Sounds,” *Proc. IEEE Workshop on App. of Sig. Proc. to Audio and Acoust.*, New Paltz, NY.

[5] J. Laroche and J.L. Meillier, “Multichannel Excitation/Filter Modeling of Percussive Sounds with Application to the Piano,” *Proc. IEEE Tr. Speech and Audio Proc.,* vol. 2, no. 2, pp. 329–344, April 1994.

[6] M. Karjalainen, V. Välimäki, and Z. Jánosy, “Towards High-Quality Sound Synthesis of the Guitar and String Instruments,” *Proc. 1993 Int. Conf. Computer Music,* pp. 56–63, Tokyo.

[7] A. Paladin and D. Rocchesso, “A Dispersive Resonator in Real-Time on MARS Workstation,” *Proc. 1992 International Computer Music Conference,* pp. 146–149, San Jose.

[8] J. O. Smith, “Music Applications of Digital Waveguides,” (A compendium containing four related papers and presentations.) CCRMA Tech. Rep. STAN–M–67, Stanford University, 1987, (415)723-4971.

[9] J. O. Smith, “Unit-Generator Implementation on the NeXT DSP Chip,” Proc. 1989 Int. Conf. Computer Music, pp. 303–306.

[10] J. O. Smith, “Physical Modeling Using Digital Waveguides,” *Computer Music Journal,* special issue: Physical Modeling of Musical Instruments, Part I, vol. 16, no. 4, pp. 74–91, Winter, 1992.

[11] J. O. Smith, “Efficient Synthesis of Stringed Musical Instruments,” *Proc. 1993 International Computer Music Conference,* pp. 64–71, Tokyo.

[12] H. Suzuki, “Model Analysis of a Hammer-String Interaction,” *J. Acoust. Soc. Amer.*, vol. 82, no. 4, pp. 1145–1151, Oct. 1987.

[13] S. A. Van Duyne and J. O. Smith, “Physical Modeling with the 2-D Digital Waveguide Mesh,” *Proc. 1993 International Computer Music Conference,* Tokyo.

[14] S. A. Van Duyne, and J. O. Smith, “A Simplified Approach to Modeling Dispersion Caused by Stiffness in Strings and Plates,” *Proc. 1994 International Computer Music Conference,* pp. 407–410, Århus.

[15] S. A. Van Duyne, J. R. Pierce, and J. O. Smith, “Traveling Wave Implementation of a Lossless Mode-Coupling Filter and the Wave Digital Hammer,” *Proc. 1994 International Computer Music Conference,* pp. 411–418, Århus.

[16] S. A. Van Duyne, and J. O. Smith, “Developments for the Commuted Piano,” (Companion paper.) *Proc. 1995 International Computer Music Conference,* Banff.

[17] G. Weinreich, “Coupled Piano Strings,” *J. Acoust. Soc. Amer.*, vol. 62, no. 6, pp. 1474–1484, Dec. 1977. Also see *Scientific American,* vol. 240, p. 94, 1979.