A Brief Review on Physics-based Piano Sound Synthesis

Tongyu Lu

April 27, 2025

Abstract

In this article, we present an overview of three major approaches to piano sound modeling: sample-based synthesis, physics-based modeling, and data-driven methods. Among these, we focus in particular on physics-based models, detailing an efficient framework that captures the interactions between the string, hammer, bridge, and soundboard. We implement a simulation based on this model and successfully generate piano-like sounds. The simulation script is available online[1]. Finally, we discuss potential future directions in physics-based piano synthesis, including the integration of deep learning techniques to enhance model realism and adaptability.

1 Backgrounds in Piano Sound Modeling

The piano is a widely used instrument in modern music, credited with its expressive pitch range and mechanical complexity. With the development of Virtual Studio Technology (VST) and Digital Audio Workstations (DAW) software, music producers are able to produce music with a single computer instead of recording the music with real instrument. As a consequence, musicians are looking for more realistic and efficient ways to create synthesized sounds. In this essay, we focus on the modeling of piano sound.

There are three major categories of piano sound synthesis: sample-based methods, physics-based methods and data-driven (machine learning) methods.

Sample-based methods use multi-sampling, recording each piano key in different dynamic levels (e.g., pianissimo to fortissimo) to capture the instrument's natural timbrel changes with playing intensity. On the user side, the recorded samples are recalled and adjusted to match the user need. Sample-based method is current most widely adopted for digital piano sound production. Successful products include Garritan CFX [1], Spectrasonics Keyscape [2], Native Instruments Noire [3] and Fluffy Scoring Piano [4]. However, a realistic sample-based piano models are typically large (can exceed 50-100 GB), which makes them cumbersome for individual musicians; on the other hand, a light sample-based piano model is normally not so realistic.

Physics-based modeling, or physical modeling synthesis, represents a fundamentally different approach from sample-based methods. Instead of relying on pre-recorded audio, this technique simulates the sound of a piano by mathematically modeling its internal physical processes: the mechanics of hammer-string interaction, string vibration, soundboard resonance, and acoustic radiation. Physics-based modeling is better at expressiveness, interactivity, and less storage as compared to sample-based methods; as a trade-off, physical modeling needs much more computation than sample-based ones. Pianoteq [5] is currently the state-of-the-art physics-based piano model, although not open-source. There also exist a few open-source projects, like a digital waveguide-based model Synthesis ToolKit in C++ (STK)[2] and a recent 3D modeling based individual project "fan455 piano synthesis"[3].

Recent years have witnessed a surge of interest in using machine learning (ML), particularly deep learning, for musical instrument modeling, including piano sound synthesis. Unlike traditional methods based on sampling or physics, ML-based synthesis relies on training models to learn mappings between musical input (e.g., MIDI or other score representations) and audio output. Google Magenta has made some breakthrough in this area, proposing the DDSP

[1] `https://github.com/lucainiaoge/piano-physics-model`

[2] `https://ccrma.stanford.edu/software/stk/`

[3] `https://github.com/fan455/fan455_piano_synthesis`

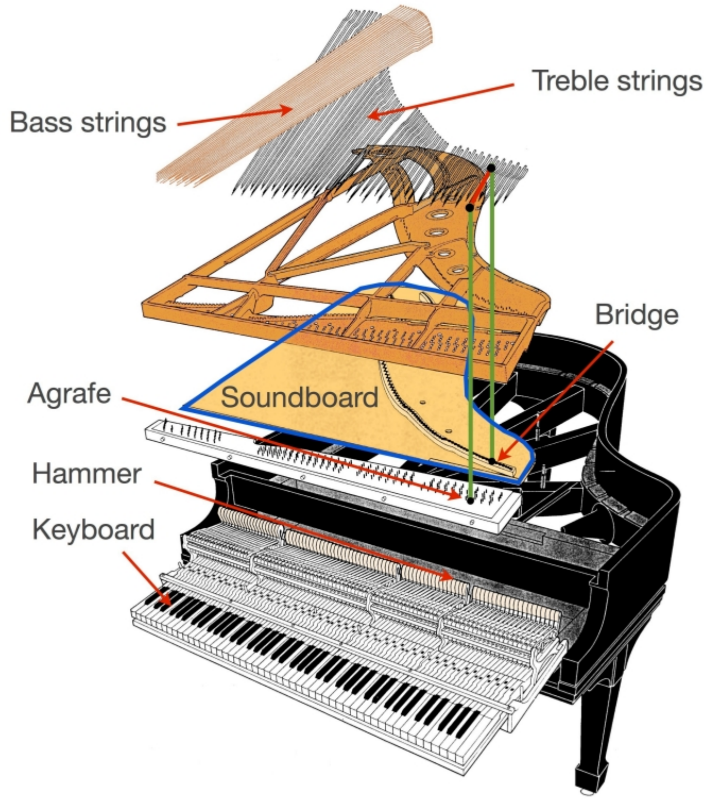


Figure 1: Exploded view of a grand piano, from [9].

(Differentiable Digital Signal Processing) [6] and NSynth (Neural Audio Synthesis) [7]. Although researches have proved the effectiveness of the deep models in instrument synthesis, there are still way to go to make the technique populate in the industry.

2 Piano Physics

In this section, we focus on physical modeling of piano, and introduce the basics of piano physics. This review is mainly based on Balázs Bank's work [8].

To model the piano, we have to first know how a piano note is activated. [9] provides a nice figure for a grand piano, which helps to understand it.

To start with, we explore the source of activation: hammer striking a piano string.

2.1 String Vibration Model

When playing a piano note, a piano string is activated and vibrate with two ends fixed. Figure 2, reproduced from a blog [4], demonstrates the string vibration in a piano.

A typical 1D string vibration model can be formulated as

$$\frac{\partial^2 y}{\partial t^2} = c^2 \frac{\partial^2 y}{\partial x^2}, \quad y(x=0,t) = y(x=L,t) = 0 \tag{1}$$

[4] `https://www.sciencelearn.org.nz/resources/2815-sound-resonance`

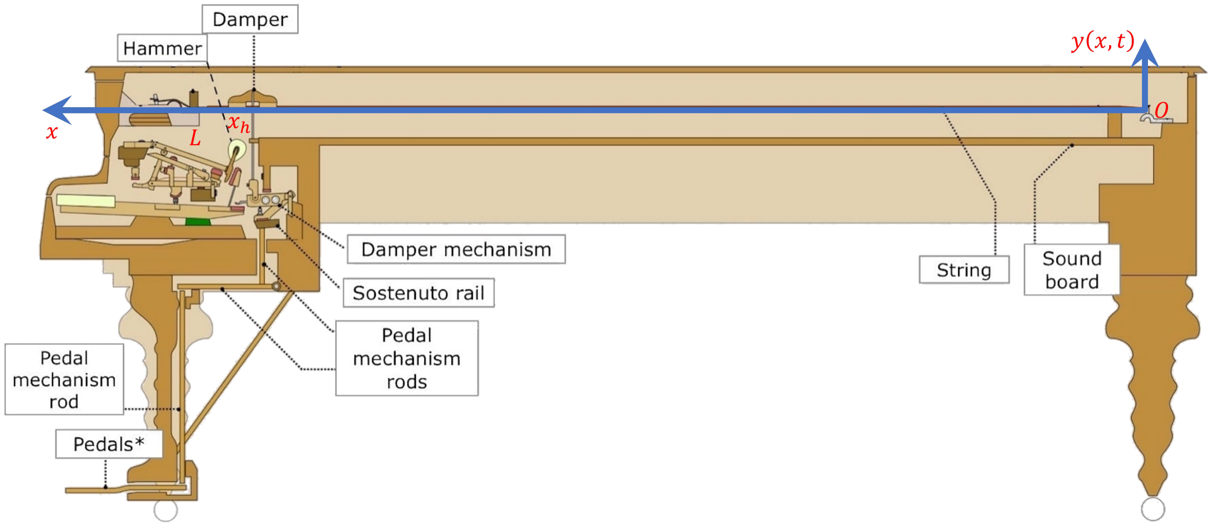


Figure 2: String-hammer-soundboard model with axis.

which can be solved through variable separation, reaching a solution

$$y(x,t) = \sum_{k=1}^{\infty} y_k(t) \sin \frac{k\pi x}{L}, \quad y_k(t) = A_k \cos \frac{k\pi c}{L} + B_k \sin \frac{k\pi c}{L}, k = 1, 2, ... \tag{2}$$

However, this solution is oversimplified for a real piano string. A more realistic version [8] can be expressed as

$$\mu \frac{\partial^2 y}{\partial t^2} - T_0 \frac{\partial^2 y}{\partial x^2} + ES\kappa^2 \frac{\partial^4 y}{\partial x^4} + 2\mu R \frac{\partial y}{\partial t} = d(x,t) \tag{3}$$

where

- μ is the density of the string (mass per unit length, typical value $3.93\mathrm{g}/0.62\mathrm{m}$ for pitch C4)
- T_0 is the tension on both sides of the string (typical value $670\mathrm{N}$ for pitch C4)
- E is the Young's modulus (measuring pressure needed to stretch something by a certain amount, typical value $2.02 \times 10^{11}\mathrm{N}/\mathrm{m}^2$ for pitch C4)
- S is the cross-section area of the string (in unit m^2, with typical value $r = 5 \times 10^{-4}\mathrm{m}$, where r is the string radius for pitch C4)
- κ is the radius of gyration (for a piano string, it can be treated as $r/2$)
- Term $ES\kappa^2 \frac{\partial^4 y}{\partial x^4}$ represents the bending force per unit length caused by the string's stiffness (which resists the normal bending as an ideal string)
- Term $2\mu R \frac{\partial y}{\partial t}$ is the frictional damping force in the piano string, which resists the motion of the string and causes energy loss over time, leading to the decay of the string's vibrations. In practice, it is frequency-dependent, and has a practical approximation [10]

  $$R_k = b_1 + b_3 2\pi f_k, k = 1, 2, ... \tag{4}$$

  where $f_k, k = 1, 2, ...$ are the mode frequencies which can be approximated by

  $$f_k \approx k f_0 \sqrt{1 + Bk^2}, k = 1, 2, ...; \quad B = \exp\left(-24.6 + 2.54 \log f_0\right) \tag{5}$$

  For pitch C4, we can set $b_1 = 0.5\mathrm{s}^{-1}$, $b_3 = 6.25 \times 10^{-9}\mathrm{s}$
- Term $d(x,t)$ is the excitation force density with unit N/m, which can be computed through the hammer model

The typical values can be found from [11], or an open-source C code by Nicholas J. Giordano[5].

To solve this more complex PDE 3, we can still assume that the solution is in form

$$y(x,t) = \sum_{k=1}^{\infty} y_k(t) \sin \frac{k\pi x}{L} \tag{6}$$

because the PDE is still constrained with boundary condition $y(x=0,t) = y(x=L,t) = 0$. To compute the modes $y_k(t)$, we can plug in Eq. 6 into PDE 3, reaching the following ODEs

$$y_k''(t) + a_{1,k} y_k'(t) + a_{0,k} y_k(t) = b_{0,k} F_k(t) \tag{7}$$

where

$$a_{1,k} = 2R_k \tag{8}$$

$$a_{0,k} = \frac{T_0}{\mu}\left(\frac{k\pi}{L}\right)^2 + \frac{ES\kappa^2}{\mu}\left(\frac{k\pi}{L}\right)^4 \tag{9}$$

$$b_{0,k} = \frac{2}{L\mu} \tag{10}$$

$$F_k(t) = \int_0^L \sin\left(\frac{k\pi x}{L}\right) d(x,t)\mathrm{d}x \tag{11}$$

To solve those ODEs, we first evaluate the impulse response with $F_k(t) = \delta(t)$. It is a linear time-invariant system equation, whose impulse response can be solved:

$$y_{\delta,k}(t) = A_k e^{-t/\tau_k} \sin(2\pi f_k t) u(t) \tag{12}$$

$$f_k = \frac{1}{2\pi}\sqrt{a_{0,k} - a_{1,k}^2/4} \tag{13}$$

$$\tau_k = 2/a_{1,k} = 1/R_k \tag{14}$$

$$A_k = \frac{b_{0,k}}{4\pi f_k} \tag{15}$$

And finally we have the ODE solution

$$y_k(t) = y_{\delta,k}(t) * F_k(t) \tag{16}$$

If we are able to express $F_k(t)$ as Eq. 11 expresses, we are capable to model the string vibration. And this term is exactly the hammer driving force mode. Next section we explore how to model the driving force.

2.2 Hammer-String-Bridge Model Basics

Let x_h be the place where hammer strikes the string (it can take a typical number like $x_h = 0.88L$), and $y_h(t)$ be the position of the hammer head.

If we ignore the hammer gravity (which is too small as compared to the string-hammer interaction), we can analyze the hammer dynamics with Newton's second law:

$$F_h(t) = -m_h y_h''(t) \tag{17}$$

where m_h is the mass of hammer, with a typical value $2.97g$

On the other hand, the driving force is related to the "compression" that the hammer felt, which can be represented by

$$\Delta y(t) = y_h(t) - y(x_h, t) \tag{18}$$

By [12], this hammer compression is useful to derive the hammer force

$$F_h(t) = K_h \left(\Delta y(t)\right)^{P_h}, \Delta y(t) \geq 0; \quad otherwise\ 0 \tag{19}$$

[5] `https://www.physics.purdue.edu/~giordano/tmp/cip/nonlinear_www.c`

where K_h is the coefficient of stiffness (typical value 4.5×10^9 for C4 pitch suggested by [11]), and P_h is the stiffness nonlinear exponent (typical value 2.5 for C4 pitch). Besides, a typical value for the initial velocity of the hammer $y'_h(t=0)$ is $2.0\mathrm{m/s}$ for C4 pitch, as suggested by Nicholas J. Giordano.

If we assume that the interaction point is infinitely small, we can express the driving force density with an impulse function:

$$d(x,t) = \delta(x - x_h) F_h(t) \tag{20}$$

and therefore the force modes are

$$F_k(t) = \int_0^L \sin\left(\frac{k\pi x}{L}\right) d(x,t)\mathrm{d}x = \sin\left(\frac{k\pi x_h}{L}\right) F_h(t) \tag{21}$$

Now, we have six (groups of) independent system variables $y_k(k=1,2,...), F_k(k=1,2,...), y_h, y(x_h,t), \Delta y, F_h$ and six (groups of) system equations 16 21 17 19 18 6. Therefore, we are able to build a feedback system and simulate the vibration.

However, this is not yet the whole picture of piano sound. As the vibration transfers from the string to the soundboard through the bridge, we should also consider the force driving the bridge to vibrate with the string. By the property of a slightly stretched steel, the bridge force can be computed with

$$F_b(t) = T_0 \left[\frac{\partial y}{\partial x}\right]_{x=0} = \frac{T_0 \pi}{L} \sum_{k=1}^{\infty} k y_k(t) \tag{22}$$

As we can already solve $y_k(t)$ with the system equations, we finally get the bridge force, which can be treated as the input of the soundboard, which amplifies the vibration and creates the piano sound.

The longitude vibration also contributes to the bridge force. For simplicity, we do not list the equations and derivations here. When the string is struck at a fortissimo level, the string motion will follow not only the transverse direction, but also a longitudinal direction. The longitudinal vibration will alter the tension force of string, producing vibration of frequency (which is called "longitudinal modal frequencies"). In addition to longitudinal modal frequencies, there is another inharmonicity coefficient in piano tones (called "phantom partials"), which is generated by the longitudinal motion of the string itself. The modeling of this effect was first explored in Balázs Bank's PhD thesis [13].

Besides, string coupling is also a salient phenomenon to consider, especially when multiple notes are activated concurrently, with the dampers lifted. [8] proposed to model the string coupling through separating strings into regions, and estimate a gain matrix to express the interactions among string regions.

2.3 Soundboard Model

As the string itself is too thin to radiate sound, the soundboard is designed as a large wooden plate which links the string through the bridge, amplifying the sound. The piano sound the audiences hear is therefore largely determined by the vibrational characteristics of the soundboard [14].

Walking through the literature, there are various ways modeling the soundboard. In [8], the filter-based soundboard modeling is discussed, which ignores the non-linear effects of the soundboard. If we treat the bridge force $F_b(t)$ as input and the final sound waveform $y_{out}(t)$ as output, the soundboard can be treated as a system $h_{sb}(t)$, and $y_{out}(t) = F_b(t) * h_{sb}(t)$. Among all digital filter design techniques, the finite impulse response (FIR) filter is most straightforward. However, as [8] pointed out, 1000–2000 tap FIR filters are needed at 44.1kHz to reach a good effect; and for reproducing the characteristic knock sound of the middle and high notes, ten thousands of taps are needed. As a consequence, [8] proposed to use fixed-pole parallel filters (IIR filters) to reduce complexity. However, as we do not have the condition to measure the impulse response of a piano soundboard, we found it hard to estimate the poles. Taking our limitations into consideration, we intend to adopt the simplified FIR method. Suppose we have discretized the system, we have $y_{out}[n] = F_b[n] * h_{sb}[n]$. With $y_{out}[n]$ and $F_b[n]$ given, we are able to estimate $h_{sb}[n]$ through linear estimation, namely

$$\mathbf{h}_{sb} = (\mathbf{F}_b^\top \mathbf{F}_b)^{-1} \mathbf{F}_b^\top \mathbf{y}_{out} \tag{23}$$

where $\mathbf{F}_b$ is the Toeplitz matrix built from $F_b[n]$.

Although a piano soundboard is relatively thin compared to its length/width, it has ribs and crown curvature, and the frequencies can get high enough that shear effects become non-negligible is we aim at creating more realistic sound [14].

Being aware of this, Ege et.al. [14] conducted finite-element simulation with bars, ribs and bridges taken into account. Chabassier et.al. [15] also developed a soundboard model by considering a piano soundboard as a bidimensional thick, orthotropic, heterogeneous, frequency dependant damped plate (as in the Reissner-Mindlin plate model), using Reissner Mindlin equations and fourth-order finite elements to conduct the soundboard simulation. Moreover, they also considered the strings-soundboard coupling at the bridge.

2.4 Discretizations and Simulation

So far, we have derived a string-hammer-bridge system. To simulate the system, we need to discretize the system and iterate in discrete time under a certain sample rate.

To start with, we consider Eq. 7, which has impulse response Eq. 12. If we set the sample rate to be f_s, and let $t_n = n/f_s$, we can get the discrete impulse response

$$y_{\delta,k}[n] = \frac{1}{f_s} A_k e^{-t_n/\tau_k} \sin(2\pi f_k t_n) u[n] \tag{24}$$

If we take z-transform on this impulse response, we can have the transfer function given $F_k[n]$ and outputting $y_k[n]$, as the relationship $y_k[n] = F_k[n] * y_{\delta,k}[n]$ suggests (the direct consequence from Eq. 16). Using the property

$$\mathcal{Z}\{e^{-aTn} \sin \omega T n\} = \frac{z^{-1} e^{-aT} \cos \omega T}{1 - 2z^{-1} e^{-aT} \cos \omega T + z^{-2} e^{-2aT}} \tag{25}$$

we can derive the string system equation $H_{str,k}(z) = \mathcal{Z}\{y_{\delta,k}[n]\}$

$$H_{str,k}(z) = \frac{b'_k z^{-1}}{1 + a'_{1,k} z^{-1} + a'_{2,k} z^{-2}} \tag{26}$$

$$b'_k = \frac{A_k}{f_s} \mathrm{Im}(p_k) \tag{27}$$

$$a'_{1,k} = -2\mathrm{Re}(p_k) \tag{28}$$

$$a'_{2,k} = |p_k|^2 \tag{29}$$

$$p_k = e^{-\frac{1}{\tau_k f_s} + j2\pi \frac{f_k}{f_s}} \tag{30}$$

During simulation, we can use the time-domain relationship derived from the system transfer function Eq. 26

$$y_k[n] = -a'_{1,k} y_k[n-1] - a'_{2,k} y_k[n-2] + b'_k F_k[n-1] \tag{31}$$

To get F_k, we should have F_h, and determine F_k through relationship expressed in Eq. 21, i.e.,

$$F_k[n] = w_{in,k} F_h[n], \quad w_{in,k} = \sin\left(\frac{k\pi x_h}{L}\right) \tag{32}$$

The system variable F_h is a result of hammer motion, and it in turn influences the hammer motion. This relationship, as specified in Eq. 19 18 6 17, can also be discretized. By Eq. 19, we have

$$F_h[n] = F(\Delta y[n]) = K_h \left(\Delta y[n]\right)^{P_h}, \Delta y[n] \geq 0; \quad otherwise\ 0 \tag{33}$$

By Eq. 18 6 17 we have

$$\Delta y[n] = y_h[n] - y_s[n] \tag{34}$$

$$y_s[n] = y(x_h, t_n) = \sum_{k=1}^{\infty} w_{out,k} y_k[n], \quad w_{out,k} = \sin\left(\frac{k\pi x_h}{L}\right) \tag{35}$$

$$y_h[n] = 2y_h[n-1] - y_h[n-2] - \frac{1}{m_h f_s^2} F_h[n-1] \tag{36}$$

What is worth noticing is that: to simulate the system, our initial condition should specify the velocity of the hammer, which is $v[0] = (y_h[0] - y_h[-1]) f_s$. If we let $y_h[-1] = 0$ and $v[0] = 2.5\mathrm{m/s}$, we are able to get the initial driving force of the whole system (assuming that all other system variables are initialized to zero).

Finally, we get the bridge force from Eq. 22 as final output, i.e.,

$$F_b[n] = T_0 \left[\frac{\partial y}{\partial x}\right]_{x=0} = \frac{T_0 \pi}{L} \sum_{k=1}^{\infty} k y_k[n] \tag{37}$$

So far, we have enough system equations to conduct a simulation for piano sound. Figure 3 provides a system diagram for the model. The next section will introduce a basic simulation experiment based on the system equations.

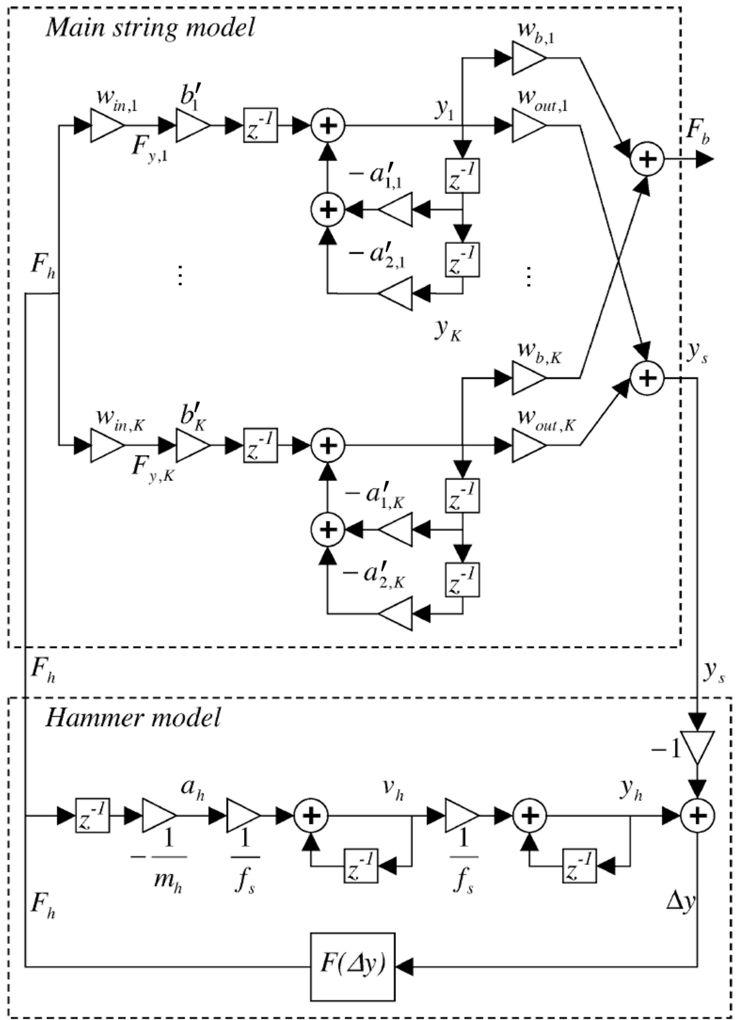
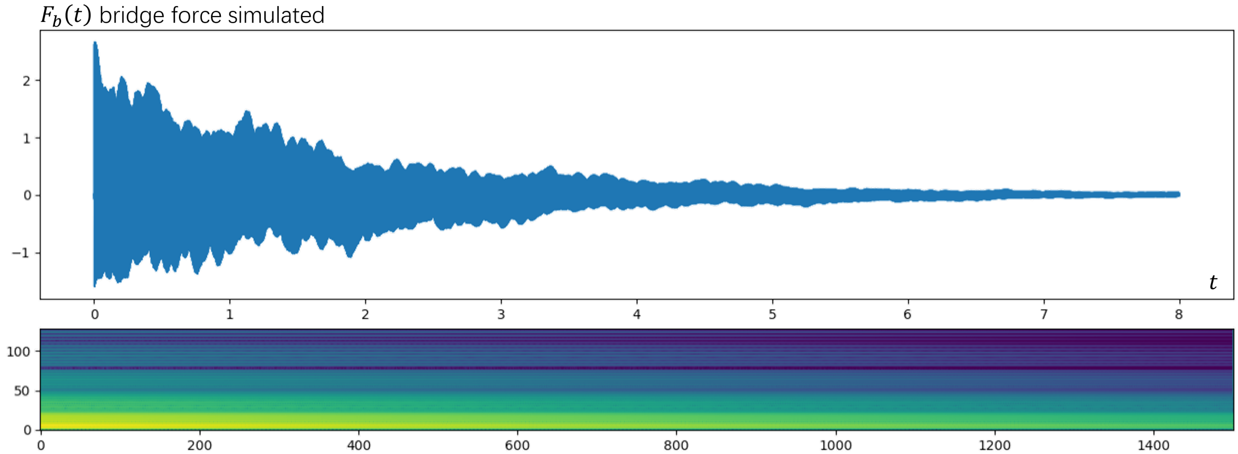


Figure 3: The string-hammer-bridge system diagram modified from the Figure 3 of [8].

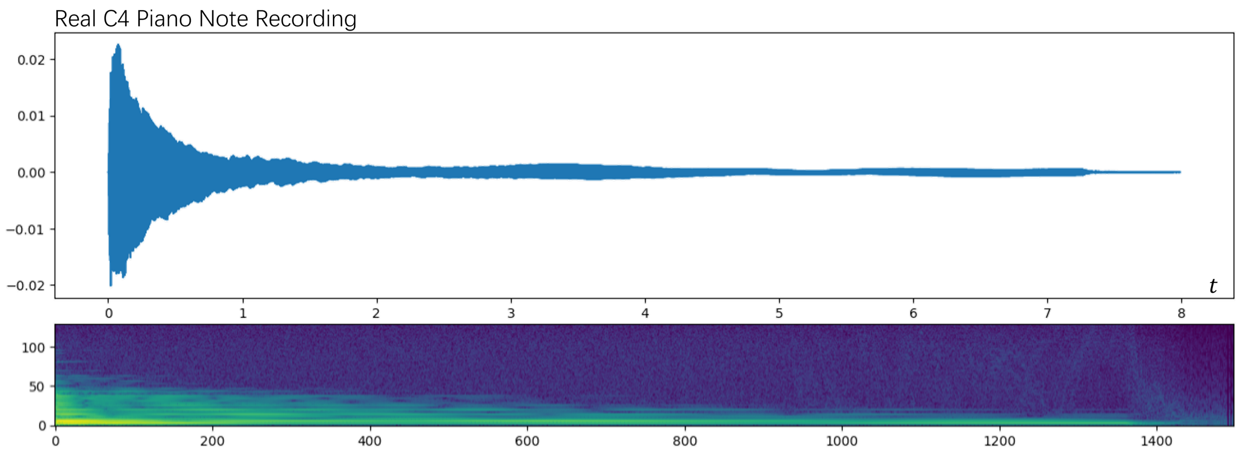
3 Simulation Experiment

We simulate the C4 pitch note with the numerical settings mentioned in the previous sections, and set the sample rate to $f_s = 24k$Hz. As there are infinitely many harmonics in the system equations, we should only consider a few harmonics approximating the system. We simulate with 37 harmonics, because higher mode frequencies exceed the half sample rate. The simulation runs Eq. 31 32 33 34 35 36 37 iteratively until reaching 8 seconds (a total of $8f_s$ time steps). Simulation results are shown in Figure 4(a).

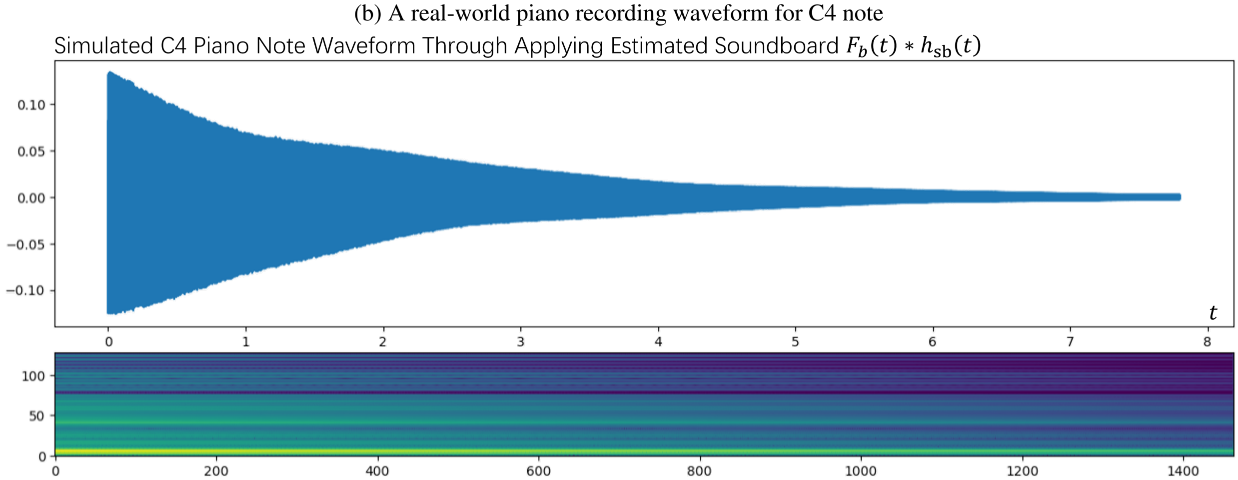
To simulate the soundboard, we define a 4800-tap FIR filter, and update the filter through fitting to a real-world piano C4 note recording, as shown in Figure 4(b). As our device does not have enough memory to run least-mean-square-estimation, we have to use stochastic gradient-decent to update the taps, with learning rate 0.01 and 10k training steps. Afterwards, we apply the estimated FIR filter on the bridge force, getting the simulated waveform output, as shown in Figure 4(c).



(a) Simulated bridge force



(b) A real-world piano recording waveform for C4 note



(c) Estimated soundboard filter applied on the simulated bridge force, getting the simulated audio output

Figure 4: Waveforms along with spectrograms of the simulated results.

Although the simulated piano note behaves like a music notes that has piano-like harmonics and faster decay at higher frequencies, there is still salient differences between the simulated note and the real-world piano note. Through listening and spectrogram comparison, we found that the simulated waveform contains a lot of high-frequency energy, while in the real-world case the high-frequency energy dampens very fast. Besides, the real-world spectrogram shows non-linear decay in harmonic modes, while the simulated spectrogram behaves like regular exponential decay. Therefore, we need to consider more non-linear effects in our system.

4 Future Directions in Physic-based Piano Synthesis

In the previous sections, we have reviewed and elaborated a piano string-hammer-bridge-soundboard model and showcased its effectiveness in simulation. However, as indicated by the simulation results, there are still a large space of improvements towards realistic synthesized sound. We identify several aspects that future researches may consider.

There is still a lack of efficient models on accurate string-bridge-soundboard coupling [9]. A most recent modeling endeavor was done by Jin Jack Tan in his Ph.D thesis [16], but not yet put into application. To step further, there are cases where the history of performance can greatly influence the resulting sound, e.g., the cases when multiple piano notes are played concurrently or sequentially. In those cases, the coupling between strings, and the historical vibration, become nonignorable in simulation. For the hammer-string model, more complex model can be designed: instead of treating the hammer-string touch as a single point, it is more realistic to treat it as a section. This brings more difficulty in system equations. Moreover, the dampers and pedaling of piano was not yet widely explored in the literature. The interaction between dampers and strings is a non-trivial effect, as the damper cannot eliminate string vibration instantly. Piano modeling in 3D was recently explored by [17], which could be a possible unified solution for the insufficiencies in the string-hammer-soundboard models. However, this endeavor is still in experiment level.

In terms of efficiency, there is still a large room of improvements. It took around 30 seconds to sample a 8 second piano note snippet, which is far from the goal running physics-based model in real-time. More advanced sampling techniques like the fixed point iteration can be applied, but the system stability is still to be explored.

Last, we propose a direction combining the model-based method with the current fast-evolving deep learning methods. Taking account of the recent advances in machine learning based synthesizing techniques, the DDSP[6] method shows a promising direction to introduce data-driven methods in refining model-based methods. Therefore, we could propose an alternative way combining the classical DSP-based piano sound modeling with machine learning: 1. simulate the bridge force as the reference inputs; 2. collect real-world piano sound data as the corresponding outputs; 3. define a neural network (e.g., WaveNet [18], Transformer [19]) that learns the relationship between the input bridge force and the resulting sound. In the dataset construction, we should provide different input bridge forces and the corresponding piano sounds in different velocity levels. To learn the coupling effect, we can even record concurrent notes.

References

[1] MakeMusic. Garritan abbey road studios cfx concert grand virtual piano. `https://www.garritan.com/products/cfx-concert-grand-virtual-piano/`.

[2] Spectrasonics keyscape. `https://www.spectrasonics.net/products/keyscape/overview.php`.

[3] Native Instruments. Noire. `https://www.native-instruments.com/en/products/komplete/keys/noire/`.

[4] Fluffy Audio. Scoring piano. `https://fluffyaudio.com/shop/scoringpiano/`.

[5] Modartt. Pianoteq. `https://www.modartt.com/pianoteq_overview`.

[6] Jesse Engel, Lamtharn (Hanoi) Hantrakul, Chenjie Gu, and Adam Roberts. Ddsp: Differentiable digital signal processing. In *International Conference on Learning Representations*, 2020.

[7] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Douglas Eck, Karen Simonyan, and Mohammad Norouzi. Neural audio synthesis of musical notes with wavenet autoencoders, 2017.

[8] Balázs Bank, Stefano Zambon, and Federico Fontana. A modal-based real-time piano synthesizer. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(4):809–821, 2010.

[9] Balázs Bank and Juliette Chabassier. Model-based digital pianos: From physics to sound synthesis. *IEEE Signal Processing Magazine*, 36(1):103–114, 2019.

[10] Antoine Chaigne and Anders Askenfelt. Numerical simulations of piano strings. i. a physical model for a struck string using finite difference methods. *Journal of The Acoustical Society of America - J ACOUST SOC AMER*, 95:1112–1118, 02 1994.

[11] Antoine Chaigne and Anders Askenfelt. Numerical simulations of piano strings. ii. comparisons with measurements and systematic exploration of some hammer-string parameters. *The Journal of the Acoustical Society of America*, 95(3):1631–1640, 03 1994.

[12] Xavier Boutillon. Model for piano hammers: Experimental determination and digital simulation. *The Journal of the Acoustical Society of America*, 83(2):746–754, 02 1988.

[13] Balazs Bank. *Physics-based Sound Synthesis of String Instruments Including Geometric Nonlinearities*. PhD thesis, 01 2006.

[14] Kerem Ege, Xavier Boutillon, and Marc Rébillat. Vibroacoustics of the piano soundboard: (non)linearity and modal properties in the low- and mid-frequency ranges. *Journal of Sound and Vibration*, 332(5):1288–1305, 2013.

[15] Juliette Chabassier, Antoine Chaigne, and Patrick Joly. Modeling and simulation of a grand piano. Research Report RR-8181, INRIA, December 2012.

[16] Jin Jack Tan. *Piano acoustics : string's double polarisation and piano source identification*. Theses, Université Paris Saclay (COmUE), November 2017.

[17] Haifan Xie. Physical modeling of piano sound, 2024.

[18] Heiga Zen Karen Simonyan Oriol Vinyals Alex Graves Nal Kalchbrenner Andrew Senior Koray Kavukcuoglu Aaron van den Oord, Sander Dieleman. Wavenet: A generative model for raw audio, 2016.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.