MASS-SPRING LATTICE PHYSICAL MODELING SYNTHESIS USING A QUANTUM ALGORITHM

Xavier Davenport
University of Illinois at Urbana-Champaign
xavierd2@illinois.edu

ABSTRACT

In this paper I describe how a recently developed quantum algorithm for simulating the dynamics of mass–spring systems can be applied in the artistic context of physical modeling synthesis, and I propose a scalable "virtual microphone" method based on amplitude amplification for extracting perceptually relevant information from the simulated system. After outlining how to synthesize sound using a one-dimensional mass–spring system with both fixed and unfixed endpoints, I discuss the scalability and implications of this approach as quantum hardware develops. Potential applications of this theoretical framework include the simulation of imperfect vibrating systems and the large-scale modeling of systems containing up to on the order of 10^{29} coupled masses in a 100 qubit system.

1. INTRODUCTION

Waveguide and modal synthesis are two of the most historically important and widely used methods of audio physical modeling. Digital waveguide methods use delay lines to model systems with high frequency losses and dispersion, and can be used to emulate the sound of strings, brass, and woodwind instruments, among others, reasonably well [1]. Modal synthesis requires knowledge of an instrument's resonant characteristics, including how the resonances dampen over time and respond to external excitation [2]. These synthesis techniques can be executed quickly enough to model instruments in real-time on modern hardware for a wide variety of resonating structures [3].

The computational efficiency of contemporary physical acoustic physical modeling techniques is achieved through approximations and simplifications of the underlying physics of vibrating systems. For example, the Karplus-Strong waveguide synthesis model assumes an ideal string that does not naturally decay in a realistic way [4]. An artificial envelope must be applied to the sound if it is intended to sound like a plucked string. There is also generally no simple way to add imperfections to the string in a waveguide model, such as one might find in the gut strings of a lute.

A more naïve approach to modeling a string involves representing the structure as a one-dimensional lattice of masses connected to each other with springs. If the string consists of only a single mass fixed via spring to two endpoints, then the plucked string has one resonant mode, the fundamental frequency, which is dependent upon the size of the mass and the strength of the springs. As more and more masses are added to the system, the number of resonant modes increases, and the string exhibits a realistic harmonic resonant spectrum. The challenge of simulating this mass-spring system is the sheer number of calculations required to describe the time evolution of the system. Every mass tugs on its neighbor, and its neighbor's neighbor, and every other mass in the system, and the same is true of every other mass! The number of computational operations required to examine the net force on each mass grows very quickly as the size of the system increases.

Recent advances in computational power and software development within the past decade have made possible the simulation of mass-spring systems containing hundreds of masses in real-time [5]. In the following, I outline a method for using quantum computers to simulate 2^{n-1} masses, where n is the number of qubits available on a quantum computer. The scaling of this simulation is exponential: with a modest number of qubits, an immense number of masses in a mass-spring system can be simulated in any number of dimensions far outpacing any classical algorithm [6]. I begin with a brief overview of the relevant concepts and mathematics from quantum computing before discussing how to map the mass-spring problem to the Schrödinger equation. I then classically simulate the quantum algorithm to find the dynamics of a plucked string with unfixed and fixed endpoints while emphasizing the simple physical intuition of this approach to modeling based on my experience as a guitarist. Next, I propose a method of using amplitude amplification to measure only the velocity of a single mass to recover an audio waveform. Finally, I discuss the long-term implications of this approach for physical modeling synthesis.

2. BASIC QUANTUM COMPUTING

A quantum computer is fundamentally different from the ordinary digital computers we interact with on a daily basis. This section provides the necessary conceptual foundation and some basic intuition for why quantum computers are a good candidate for large-scale mass-spring simulation. For a deeper understanding of quantum computation, I recommend this introduction to quantum computing for computer musicians [7], this introduction to quantum computing for non physicists [8], and the seminal text on quantum computing by Nielsen and Chuang [9].

The property of quantum systems most relevant to the calculations described in this paper is superposition. There is a quantum version of the bit called a qubit that can exist in a superposition of the states $|0\rangle$ and $|1\rangle$. The binary state 0 is represented by $|0\rangle$, the binary state 1 is represented by $|1\rangle$, and they can each be represented using matrices in the following way.

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix} \qquad |1\rangle = \begin{pmatrix}0\\1\end{pmatrix} \tag{1}$$

An arbitrary superimposed state $|\psi\rangle$ can be written as a linear combination of these two states in the form

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix}\alpha\\\beta\end{pmatrix}. \tag{2}$$

When the qubit is measured at the end of a calculation, the state collapses into either $|0\rangle$ with probability α^2 or $|1\rangle$ with probability β^2. The normalization condition $|\alpha|^2+|\beta|^2 = 1$ ensures these probabilities sum to one.

The computational power of quantum systems becomes apparent when qubits are combined to form multi-qubit states. For two arbitrary states $|\psi_1\rangle$ and $|\psi_2\rangle$, their combined state can be made using a tensor product in the following way.

$$|\psi_1\rangle \otimes |\psi_2\rangle = \begin{pmatrix}\alpha_1\\\beta_1\end{pmatrix} \otimes \begin{pmatrix}\alpha_2\\\beta_2\end{pmatrix} = \begin{pmatrix}\alpha_1\alpha_2\\\alpha_1\beta_2\\\beta_1\alpha_2\\\beta_1\beta_2\end{pmatrix} \tag{3}$$

This sort of combined state is often written in shorthand to omit the tensor product symbol as $|\psi_1\rangle \otimes |\psi_2\rangle = |\psi_1\psi_2\rangle$. For every qubit added, the number of possible output states doubles. In other words, the dimension of the state vector grows exponentially with the number of qubits. For example, if I am working with three qubits, then there are eight possible outcomes when I measure the system (e.g., $|000\rangle$, $|001\rangle$, $|010\rangle$, ... etc.). The simplest way to reconstruct the probability distribution across these states requires many measurements of identically prepared systems. A large part of research surrounding quantum algorithms is (1) developing clever methods of creating a superimposed state then (2) efficiently extracting information from it. As I discuss in the next section, there happens to be an effective way to describe the energies of masses in a mass-spring system where each possible measurement outcome corresponds to a mass velocity. Later in this paper, I discuss how to use amplitude amplification to efficiently extract information from the system.

3. MASS-SPRING HAMILTONIAN SIMULATION

In a mass spring-system, the Newtonian force $F = ma$ is equal to the restoring force defined in Hooke's law, $F = -kx$. The motion of a single mass connected to a single spring fixed at the opposite end can be described using

$$ma = -kx, \tag{4}$$

where m is mass, a is acceleration of the mass, k is a spring constant, and x is the displacement of the mass from equilibrium. For a lattice of N coupled masses, the equations of motion generalize to a system of differential equations. The net force on mass i between any pair of masses mediated using a spring can be described with the equation

$$m_i a_i = -\sum_j k_{i,j} x_j, \tag{5}$$

where i and j are mass indices.

To simulate this system on a quantum computer, the above classical equations of motion must be mapped to the time dependent Schrödinger equation, given by

$$\frac{d}{dt}|\psi(t)\rangle = -iH|\psi(t)\rangle, \tag{6}$$

where $|\psi(t)\rangle$ is the state of the system at a given time and H is the Hamiltonian. As recently shown by Li, an effective mapping is achieved by finding Q such that

$$Q^T Q = \mathbf{k}, \tag{7}$$

where **k** is a matrix representing all spring coefficients, and Q^T is the transposed Q matrix [6]. The state of the system is given by

$$|\psi\rangle = \frac{1}{\sqrt{2\mathbf{E}}}\begin{bmatrix}\sqrt{M}\mathbf{v}\\ iQ\mathbf{x}\end{bmatrix}, \tag{8}$$

where **E** describes the total energy of the system, **v** is the velocity of the masses, and M is a diagonal mass matrix. The Hamiltonian is therefore given by

$$H = -\begin{bmatrix}0 & \sqrt{M}Q^T\\ Q\sqrt{M} & 0\end{bmatrix}. \tag{9}$$

Finding the matrix Q is nontrivial, however, Li gives a matrix describing a one-dimensional system with nearest-neighbor and next-nearest neighbor interactions of the form

$$Q = \begin{pmatrix} q_0 & 0 & 0 & \dots & 0\\ q_1 & q_0 & 0 & \dots & 0\\ q_2 & q_1 & q_0 & \dots & 0\\ 0 & q_2 & q_1 & \ddots & 0\\ \vdots & \ddots & \ddots & \ddots & q_0 \end{pmatrix}, \tag{10}$$

where in a mass-spring system, q_0 describes the self restoring force for a given mass, q_1 describes the nearest-neighbor interactions, and q_2 describes the next-nearest-neighbor interactions. Each diagonal band of q_0, q_1, and q_2 can, in principle, contain distinct values for each entry, enabling the modeling of inhomogeneous systems.

4. PLUCKED STRING

In the following, I apply this framework to a plucked string modeled as a one-dimensional lattice of N masses with nearest and next-nearest neighbor couplings. To simulate a pluck, the state of the system is initialized such that the masses are displaced in the shape of a triangle; one mass is maximally displaced, and other masses are linearly aligned towards the fixed endpoints. I define the location of the pluck (i.e. the mass with maximal initial displacement) as x_0.

The simplest simulation case is one where the endpoints of the string are not fixed, as depicted in Figure 1(b). The result is that energy never leaves the system, and the recovered sound is inharmonic due to the lack of energy dissipation.

To more realistically simulate a plucked string, the endpoints must be fixed, as depicted in Figure 1(a). Because there is no way to evolve different masses under multiple distinct equations of motion using this framework, the string must be simulated as un-fixed, then manually adjusted at each timestep. Manually altering the endpoints to ensure their velocity is always zero therefore removes energy from the string, creating an envelope over the audio waveform. Such a constraint requires that the state of the system is re-normalized after the calculation at each timestep, adding significant computational overhead. The consequence of continually renormalizing the system, however, is that the entire statevector of the system must be precisely known for every timestep, requiring a large number of measurements. A way to work around this issue is by adding damping to the system and ensuring that the endpoint masses have very high damping coefficients. This procedure is beyond the scope of this paper, but it is discussed in the Future Work section.

The velocity of each mass is averaged to produce a single sample at a given timestamp. **Figure 2** shows the velocity of each mass in a sixteen mass system at $t = 0$ when the string is released from the plucked position, as well as snapshots shortly after the release. The string is plucked at the 5th mass. The non-uniform velocity represented at timestamps after the pluck result from assigning each mass a slightly different weight, modeling an imperfect string.

To add another level of realism to the simulation, the string should be additionally coupled to dissipative elements such as the body of an instrument or air surrounding the instrument. If sufficient dissipative elements were to be simulated along with the vibrating string, the energy from the initial string pluck would dissipate throughout the rest of the system, allowing the string vibration amplitude to decay. This requires a new mapping from the classical equation of motion for a damped mass-spring system to the Schrödinger equation, a task beyond the scope of this paper.

4.1 Simulation Procedure

The procedure to simulate a one-dimensional vibrating string without dissipation is as follows:

1. Define the static parameters of the system. This includes the number of masses N, the mass values m_i, the tension coefficients γ, the time step (ex. 1/44,100), and the total desired audio sample size.
2. Define the initial displacement of the string. In a physical sense, define how the string is pulled from equilibrium just before it is plucked. As an example, the maximally displaced mass x_0 can be displaced by an amplitude $A = 1$. In practice, the masses can be initialized to any collection of displacements.
3. Construct q_0, q_1, and q_2, corresponding to the self restoring force, the interaction strength between a mass

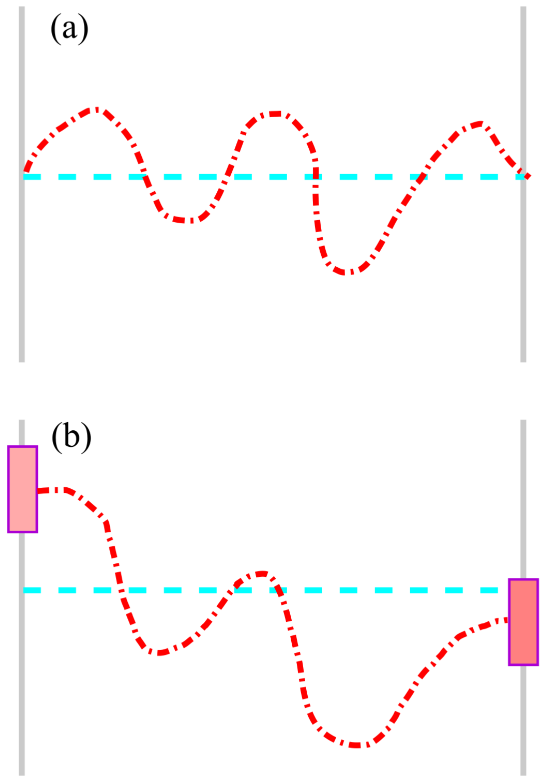

Figure 1. Sketch of a plucked string with fixed (a) and unfixed (b) endpoints. In the fixed case (a), the masses at either extreme of the string are assigned zero velocity. In the unfixed case (b), there is no such restriction. In both the fixed and unfixed cases, each mass in the mass-spring lattice can be thought of as only having freedom of motion along the y-axis. In an experimental setup, an unfixed vibrating string can be approximated by fixing the endpoints of the string to frictionless carts allowed to move along the y-axis as represented by red boxes in (b).

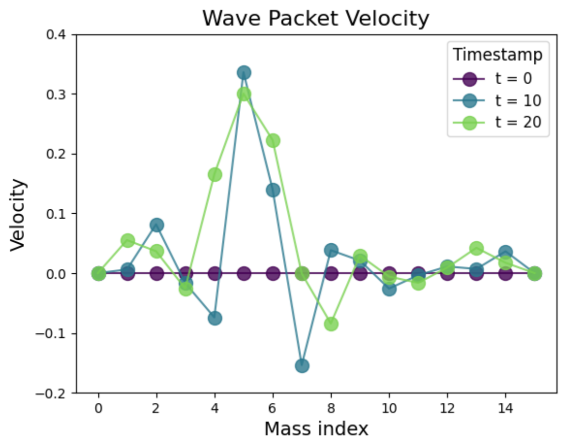

Figure 2. Plot of mass velocity in a sixteen mass mass-spring system with fixed endpoints. The system begins at rest when $t = 0$. Mass number 5 is released, causing it to have great positive velocity at $t = 10$, and the energy wave disperses slightly in either direction by $t = 20$.

and its nearest neighbor, and a mass and its next nearest neighbor. For example,

$$q_0 = \gamma \frac{1 + 1/\sqrt{3}}{2}, \tag{11}$$

$$q_1 = -\gamma, \tag{12}$$

$$\text{and} \quad q_2 = \gamma \frac{1 - 1/\sqrt{3}}{2}, \tag{13}$$

where γ is the tension coefficient defined in the previous step, which can be used to change the resonant frequencies of the string. The coefficient γ can be thought of as a tuning peg on a violin.

4. Construct the matrices Q, H, and the initial input state $|\psi(0)\rangle$.

5. Compile the quantum time-evolution circuit using the input state and Hamiltonian operator. Note that the compilation of the Hamiltonian into a manageable number of qubit logic gates is an active area of research, and the subtleties of this process are beyond the scope of this paper.

6. Execute the quantum circuit and measure the output state.

7. Repeat steps 5 and 6 enough times to adequately reconstruct the state of the system. The number of measurements required for the desired level of accuracy scales linearly with the number of masses. The resulting distribution represents the velocity of each mass at a given timestamp.

8. Sum and normalize the measured velocities to find the net amplitude and save the value as an audio sample.

9. Repeat steps 5 through 8, plugging in a new time value in the time-evolution operator for each sample.

4.2 Fixed Endpoints

If the ends of the string are fixed such that the velocity of the first and Nth masses are zero, the quantum state must be renormalized after each timestep. The updated state then becomes the input for the next timestep. This has significant ramifications in terms of measurement complexity. For an unfixed string, only the average velocity of a mass-spring system is required to construct an audio sample, requiring relatively few measurements. In contrast, simulating a fixed-endpoint string requires reconstructing the wavefunction to very high resolution so as not to distort the physics at every timestep. The number of measurements needed to fully reconstruct the wavefunction scales linearly with the number of masses in the system, negating much of the quantum computational advantage. Advantage can be recovered using a amplitude estimation algorithm or with careful application of amplitude amplification as detailed in Section 5.

One additional step inserted after step 7 is required to calculate the dynamics of a one-dimensional string with fixed endpoints: Set the velocity of the first and last mass to zero, and renormalize the state vector. The renormalized state vector is the new input state for step 5.

4.3 Simulation Results

The following analysis focuses on the fixed endpoint plucked string as it produces the harmonic spectrum and amplitude

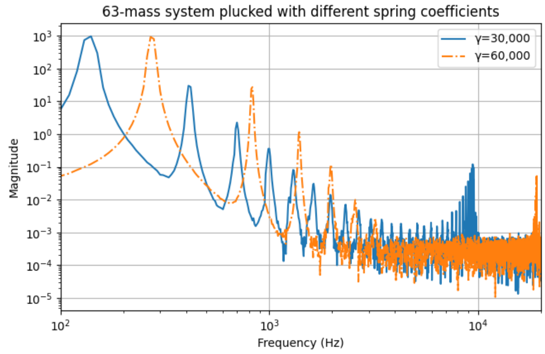

Figure 3. Plot of frequency spectra for mass-spring systems with 63 masses and fixed endpoints 0.05 seconds after the pluck for two different tension coefficients γ. Increasing γ increases the pitch of the simulated string.

decay characteristics of a real string. [1] The most significant difference between the fixed and unfixed cases is that energy never leaves the unfixed string. Initial conditions such as pluck location, irregular masses, etc. result in a complex inharmonic spectra that never decay. This suggests that the unfixed string may be a new and idiomatic form of quantum tone synthesis.

The following plots and discussion are intended to show that the alterable parameters in a mass-spring system are useful in simulating subtle aspects of string vibration in an intuitive way. My approach to varying parameters and exploring the characteristics of the system is guided by my own background as a guitarist. Just as I twist the tuning peg to raise or lower the tension of a guitar string, I can vary the tension parameter γ to shift the frequency spectrum of the plucked string. **Figure 3** shows the frequency spectra of a string with 63 masses for different tension coefficients γ. As the tension grows, so does the fundamental frequency and the harmonic series above it. Due to the small number of masses used to simulate the string, there is an increase in high frequency harmonics approaching 10^4 Hz with γ set to 30,000 which decays after letting the string ring for a short duration.

As a guitarist, I also think about the location from which I pluck the string and how that may affect the brightness of the sound. Plucking closer to where the string is fixed to the bridge of the guitar *sul ponticello* is generally much brighter due to an increased volume of the upper partials compared to a *sul tasto* pluck near the fingerboard of the guitar where upper partials are suppressed. **Figure 4** shows a plucked string plucked from a *ponticello* position at mass 62 next to the fixed endpoint and another string plucked at its midpoint. The resulting spectra show a relative boost in the amplitude of the partials of the *ponticello* pluck near the bridge compared to the *tasto* pluck.

Though I have little control over this in real time as a performer, I have certainly played a guitar strung with old and damaged strings. To emulate the effect of damaging a

[1] The code used to simulate both fixed and unfixed strings using a quantum algorithm, alongside several hundred audio samples, can be found at `https://github.com/Xavman42/TENOR26_Mass-Spring_lattice_plucked_string`.

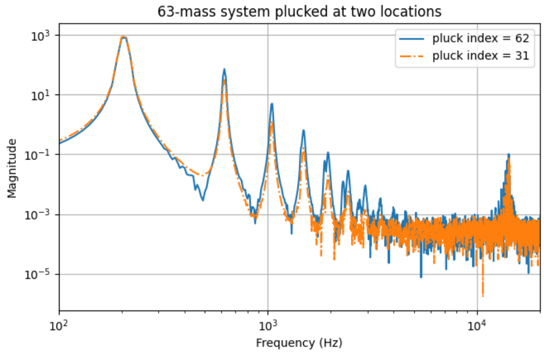

Figure 4. Plot of frequency spectra for mass-spring systems with 63 masses and fixed endpoints 0.05 seconds after the pluck for two different pluck locations. Plucking at the 62nd mass is akin to plucking a string close to the bridge (i.e. *sul ponticello*), wheras plucking at the 31st mass is akin to plucking from the middle of the string. Note how the upper partials are slightly louder in the *ponticello* pluck than the mid-point pluck.

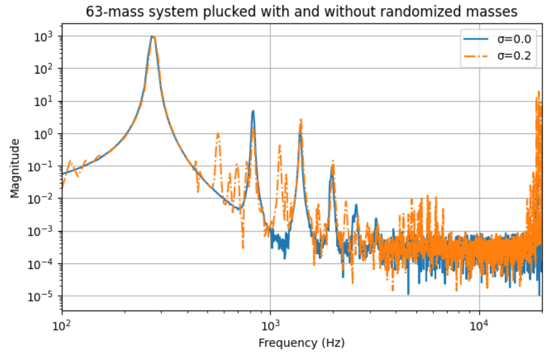

Figure 5. Plot of frequency spectra for mass-spring systems with 63 masses and fixed endpoints 0.05 seconds after the pluck for two different mass distributions. $\sigma = 0.0$ represents a system where every mass is identical, while $\sigma = 0.2$ represents a system where masses vary in size from 80% to 120% of the default mass. Note that additional inharmonic tones are present in the spectra when the masses are not uniform.

string, I can vary each mass in the system to make it non-uniform. In **Figure 5**, I plot the spectra of a mass-spring system where every mass is identical and compare that with another string where each mass varies by up to 20%. While the fundamental frequency is largely unchanged, some of the upper partials are amplified, and others are attenuated. More strikingly, distinct inharmonic components appear in the attack transient, though they quickly decay after a short time. Perceptually the string sounds more dead, as though the string has been played a lot and for a long time, both damaging the string and adding layers of skin and oil to the surface.

5. VIRTUAL MICROPHONE

The standard method to extract the entire wavefunction is to repeatedly prepare, run, and measure the final state of a quantum circuit to recover $|\psi(t)\rangle$. For the fixed-endpoint simulation described above, this full reconstruction of the state is required at every timestep because the updated state must be precisely known to serve as the input for the next sample calculation. The number of measurements required to reconstruct a distribution up to an arbitrary error scales linearly as the number of masses increases. Considering that the number of masses which can be simulated is given by 2^{n-1} where n is the number of qubits in the system, it is worth investigating if there is some alternative way to extract information from the system.

To limit the number of measurements required to recover useful information from the system, I propose a "virtual microphone" technique based on quantum amplitude amplification. The approach is inspired by how membrane microphones deflect in response to local air pressure variations and convert mechanical energy into an electrical signal. In short, amplitude amplification is a method of increasing the likelihood of measuring one or more states at the end of a quantum calculation. For example, an arbitrary encoded state is shown in **Figure 6(a)**, and the state after amplitude amplification is shown in **Figure 6(b)**. The relative magnitude between the target states is preserved, and the likelihood of measuring any other state is suppressed.

Acoustically, humans do not perceive sound as the velocity of every atom in the vicinity. Rather, we detect local pressure variations using the tympanic membrane in the ear. Similarly, a guitar pickup can be thought of as an electromagnetic microphone which measures only the nearest portion of a vibrating string. An idealized guitar pickup for the plucked string described in this paper could be modeled as the velocity over time of only a single mass in the string.

Extracting the specific amplitude of a single mass via amplitude amplification requires defining the target state relative to some reference. The solution I propose is to simulate two identical but classically separated mass-spring systems within the same quantum circuit which differ only by an overall sign in their initial displacement. Both the ordinary and inverted systems evolve identically in time but with opposite velocities for every mass. The joint quantum state of both strings maintain anti-symmetry throughout for any timestamp because they obey the same physics.

Doubling the computational size of the system and accounting for the inverted string can be done at the cost of only a single additional qubit. Amplitude amplification is then applied to a pair of basis states, one mass in the primary system and the other corresponding mass in the inverted reference system. After amplification, the probability ratio of measuring these two states is equal to the square of the target mass's velocity. Regardless of the number of masses N, estimating a probability ratio between two states requires a number of measurements dependent only on the desired precision. This technique can also be generalized to amplify more than a single pair of states, effectively emulating a microphone with a wider diaphragm.

The amplitude amplification algorithm would be appended at the end of the quantum circuit generated in step 5 in

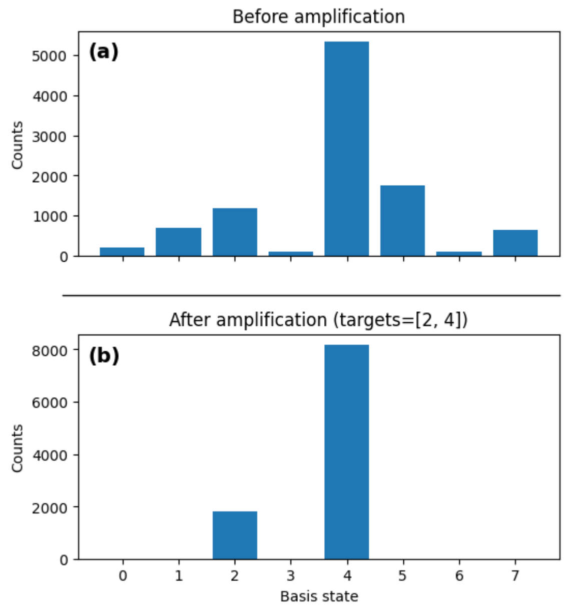

Figure 6. Plots showing the before **(a)** and after **(b)** of an amplitude amplification procedure targeting basis states 2 and 4. Note the change of scale in the y-axis. In both cases, the system was measured 10,000 times.

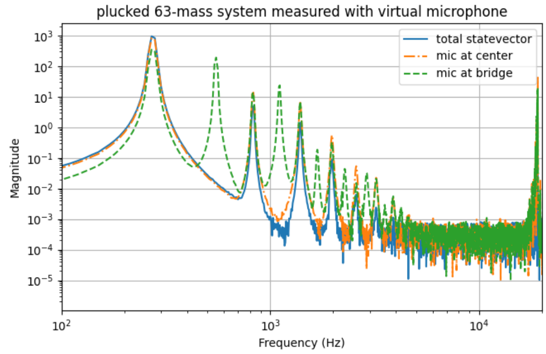

Figure 7. Plot of frequency spectra for mass-spring systems with 63 masses and fixed endpoints 0.05 seconds after the pluck for three measurement methods. The total statevector is the measurement method used in all previous graphs where the velocity of all masses in the system are summed and normalized to create an audio sample. The "mic at bridge" spectrum was created from measurements of just a single mass, close to the end of the string. The "mic at center" spectrum was similarly created using a mass at the center of the string. Note how measuring a mass close to the center of the string provides a reasonable approximation of the entire state vector. Measuring a mass closer to the fixed endpoint instead captures more partials, producing a brighter sound on the attack akin to the bridge pickup on an electric guitar.

the previous section. Simulating this quantum amplitude amplification is computationally prohibitive for mass-spring systems large enough to be sonically interesting. To validate the sonic implications of this virtual microphone idea, I can instead extract the velocity information of a single mass from the state vector simulations in the previous section.

The frequency characteristics of a plucked string for two virtual microphone locations in a fixed-endpoint system are plotted in **Figure 7**. Measuring a mass at the center of the string generates a waveform exceptionally similar to the average mass velocity from the total statevector measurement. Measuring a mass close to the fixed endpoint (i.e. close to the bridge on a guitar) brings out additional overtones which decay quickly after the initial transient response right after the string is released.

6. DISCUSSION

Mass-spring systems offer an intuitive method for audio physical modeling, but scale exceptionally poorly using any known digital computing algorithm. This paper points out that quantum Hamiltonian simulation can dramatically increase the calculation speed for this system and efficiently model mass-spring lattice at classically intractable scales. I have also proposed a "virtual microphone" technique based on amplitude amplification for efficiently extracting localized audio signals from a quantum state which ensures the same number of system measurements are required to extract useful audio information for any size of mass-spring system.

It is important to emphasize that this quantum simulation framework for mass-spring systems is scalable to any number of spatial dimensions. While circuit depth grows in a multiplicative way as the number of spatial dimensions increases, the exponential scaling in the number of simulated masses remains, [6] meaning that modeling a large-scale three dimensional mass-spring system is computationally feasible. Current noisy intermediate-scale quantum processors host over 100 qubits and can execute thousands of gates in a single circuit, though their noise limits output fidelity [10]. Rapid progression towards fault-tolerant quantum computation suggests that it is not unreasonable to expect the acoustic simulations outlined in this paper to be accessible on actual quantum hardware within a few years.

With a mere 100 qubits, on the order of 10^{29} masses in a mass-spring lattice system could be simulated using the procedure outlined in this paper, in any number of spatial dimensions. The primary caveat to this statement is that only so many operations can be made upon modern qubits before they become subject to noise and decoherence, destroying the data, but hardware is rapidly improving and hardware may one day no longer be a significant restriction. To demonstrate just how large that number is, consider a trumpet. A trumpet weighs approximately 1.2 kilograms, and is made mostly of brass. Yellow brass has the chemical formula CuZn33 with a molar mass of 2.221 kilograms, so a trumpet contains approximately 0.54 moles of brass, equivalent to roughly 3.24×10^{23} atoms. If I also model the air surrounding the brass as a system of weakly coupled masses and springs, then I could estimate that the total system of a trumpet and the air around it could be simulated using approximately 10^{24} masses. In other words, a 100 qubit computer with sufficient fidelity and gate depth should be able to simulate a system of 10 billion trumpets and the air surrounding the instruments at the atomic level.

Physically modeling instruments using this mass-spring lattice approach requires knowledge about the material makeup of objects. Even the most pure brass alloys have some imperfections which could be reasonably modeled using this method. Simulated wood and other organic materials used to make instruments would also resonate in a significantly more realistic fashion if material irregularities could be simulated. For this to be a realistic synthesis technique, much research is needed around recreating the resonant characteristics of different materials using masses and spring coefficients.

Perhaps the most significant step towards increasing the realism and practicality of this procedure would be the inclusion of damping coefficients in the mass-spring system [11]. Modeling velocity dependent dissipative forces such as viscous friction would naturally produce a sound envelope in the waveform. Further research is required to find the correct mapping from the classical damped mass-spring system to the Schrödinger equation following Li [6]. If damping can be accounted for in the unitary time evolution of the quantum circuit, then there would be no need to renormalize the state vector describing a string fixed at both ends because a given mass can be effectively fixed by over-damping its interactions.

7. CONCLUSION

I have outlined how to simulate the evolution of a one dimensional mass-spring system after it was plucked like a guitar string using a quantum computer. I have also proposed a method of effectively extracting information from the system using a "virtual microphone" technique by duplicating the classical system and applying amplitude amplification to the target masses.

The significance of this procedure lies not in its immediate practicality, rather it lies in its promise to afford us a new way to simulate large and complicated acoustic systems in a precise way without the need of approximations and assumptions made in wave-guide and other synthesis techniques. To accurately simulate sound traveling through materials and transitioning between objects and the air requires understanding how to accurately represent the acoustical behavior of the materials using mass-spring lattices. As the theoretical foundation of this synthesis technique is made more robust and as quantum hardware becomes more accessible, mass-spring systems may prove useful both for the synthesis of realistic sounds and for improving our understanding of how sound propagates through the world around us.

8. REFERENCES

[1] J. O. Smith, "Physical modeling using digital waveguides," *Computer Music Journal*, vol. 16, no. 4, pp. 74–91, 1992. [Online]. Available: http://www.jstor.org/stable/3680470

[2] J. M. Adrien, *Representations of Musical Signals*. MIT Press, 1991, ch. The missing link: Modal synthesis, p. 269–297.

[3] C. Bruyns, "Modal synthesis for arbitrarily shaped objects," *Computer Music Journal*, vol. 30, no. 3, pp. 22–37, 2006. [Online]. Available: http://www.jstor.org/stable/4617941

[4] K. Karplus and A. Strong, "Digital synthesis of plucked-string and drum timbres," *Computer Music Journal*, vol. 7, no. 2, pp. 43–55, 1983. [Online]. Available: http://www.jstor.org/stable/3680062

[5] J. Villeneuve and J. Leonard, "Mass-interaction physical models for sound ans multi-sensory creation : Starting anew," 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:204780346

[6] X. Li, "Exponential quantum speedup for simulating classical lattice dynamics," *Phys. Rev. Lett.*, vol. 135, p. 080602, Aug 2025. [Online]. Available: https://link.aps.org/doi/10.1103/z2jq-1rxp

[7] X. Davenport, "Single sample per qubit audio signal processing: A crash course in quantum computing for computer musicians," *Emille*, vol. 23, pp. 31–44, 2025.

[8] C. Bernhardt, *Quantum computing for everyone*. Cambridge: The MIT Press, 2019 - 2019.

[9] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*, 10th ed. Cambridge ;: Cambridge University Press, 2000. [Online]. Available: http://catdir.loc.gov/catdir/samples/cam031/00028911.html

[10] "Ibm quantum computing — hardware and roadmap," https://www.ibm.com/quantum/hardware, 2026, accessed: 2026-01-04.

[11] D. Morgan and S. Qiao, "Analysis of damped mass-spring systems for sound synthesis," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2009, no. 1, p. 947823, Jun 2009. [Online]. Available: https://doi.org/10.1155/2009/947823