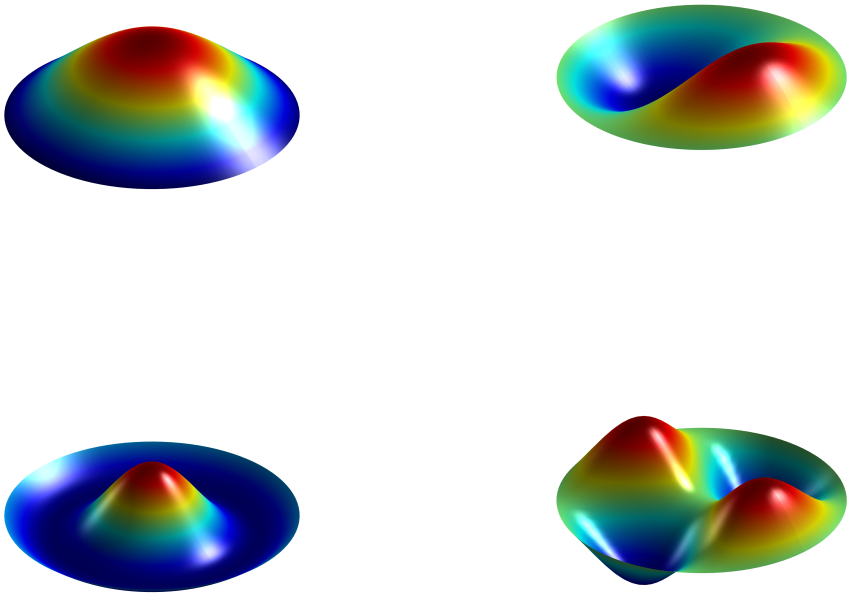
Physical Modeling of a Percussion Musical Instrument Using Mass-Spring Network System

Master's thesis in Sound and Vibration

ALBIN DUFVENIUS ESPING & SAM SOLIMAN



DIVISION OF APPLIED ACOUSTICS
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Physical Modeling of a Percussion Musical Instrument Using Mass-Spring Network System

Albin Dufvenius Esping
albindu@chalmers.se

Sam Soliman
samsol@chalmers.se

Master Thesis ACEX30

June 24, 2025

CHALMERS

Division of Applied Acoustics
Chalmers University of Technology
Gothenburg, Sweden 2025

Abstract

This work investigates physical modeling techniques for percussion instruments, with a focus on developing a practical and computationally efficient drum sound synthesis. A comparison was made between four major physical modeling approaches, which are Finite Difference Time Domain (FDTD), Delay Line, Modal Synthesis, and Mass-Spring Network. The mass-spring approach emerged as the most appropriate method, balancing feasibility with computational efficiency. Thus, two frameworks were implemented: a detailed offline model with Cartesian-grid representation, integrating complex membrane physics in the model; and an optimized polar-coordinate real-time version, which is implemented as both a VST plugin and VCV Rack module. Spectral and modal analyses confirm that the models mentioned above reproduce successfully the characteristic resonant frequencies predicted by Bessel function theory. When comparing with recordings of acoustic drums, the model cannot capture all the spectral nuances of physical drums. However, it successfully reproduces the perceptually significant modal patterns that define different drum types. The real-time implementation maintains these essential characteristics while introducing performance-responsive behaviors that are not typically achievable with sample-based instruments. This research contributes to both the theoretical understanding of percussion acoustics and practical applications in music technology, offering a methodology that fills the gap between physical accuracy and computational practicality for drum sound synthesis.

Acknowledgments

This thesis work has been done in collaboration with Elektron Music Machines where we would like to give a special thanks to:

Mattias Rickardsson
Nikolaj Andersson
Jimmy Myhrman

Who have helped us during the whole processes of this thesis work with great insights into synthesis techniques, digital signal processing, and sound synthesis methodologies, being essential to this project's success. The deep expertise they shared in modeling implementation and real-time audio processing helped bridge the gap between theoretical concepts and practical implementation, making it possible to transform ideas into a functional real-time plugin.

Contents

Glossary

Term	Definition
CFL condition	Courant-Friedrichs-Lewy condition; a mathematical stability criterion for numerical simulations that restricts the time step based on spatial grid spacing and wave speed. If violated, the simulation becomes unstable.
Dirac delta function	A mathematical function that is zero everywhere except at a single point, with an integral of one over its entire domain. In physical modeling, it represents an idealized impulse like an instantaneous strike on a membrane.
Euler integration	A first-order numerical method for solving differential equations that updates the state of a system based on its current rate of change.
MIDI	Musical Instrument Digital Interface; a technical standard for communication between electronic musical instruments and computers. In this thesis, MIDI signals provide velocity information.
Plugin	A software component that adds specific features to an existing computer program. In audio contexts, plugins like VSTs extend digital audio workstations with virtual instruments or effects processors.
Real-time processing	Computation performed quickly enough that results are available as needed for seamless user interaction. In audio, this means processing must complete within strict time constraints to avoid audible artifacts.
DAW	Digital Audio Workstation; a software application used for recording, editing, mixing, and producing audio files. DAWs serve as the host environment for the VST plugin implementation, providing the interface between the physical model and music production workflow.
Singularity	A point where a mathematical function or physical quantity becomes infinite or undefined. In polar coordinate systems, the center point (r=0) represents a singularity.
Verlet integration	A second-order numerical method for integrating equations of motion. This method stores previous and current positions to calculate the next state.

Term	Definition
VST	Virtual Studio Technology; an audio plugin interface standard developed by Steinberg that allows third-party instruments and effects to be used within digital audio workstations.
Offline model	A simulation model implemented in this work to represent all the computations needed, regardless of how long it takes for the calculations in the model. The purpose is to obtain a high-resolution physical simulation without being limited by real-time implementation constraints.
Real Time model	A drum simulation model optimized in this thesis for immediate sound generation with minimal latency, using physics and efficient algorithms to operate within strict computational constraints. It is presented as both a VST plugin for DAWs and a module for VCV Rack, prioritizing performance.
Eurorack	A modular synthesizer format standardized by Doepfer in the 1990's [1]. In this thesis, VCV Rack software simulates the Eurorack environment, where modules can interact with other synthesis components through standardized voltage control signals and audio pathways.
Modal Frequency	The natural frequency of an oscillating mode. In circular membranes, it is determined by Bessel roots.
Mode Shape	A spatial vibration pattern of a structure at a specific modal frequency.
Laplacian (∇^2)	A differential operator that represent the spatial curvature of a field such as displacement or velocity. It is used to calculate curvature and damping.
Gaussian Envelope	A bell-shaped function applied over time to shape an excitation signal, simulating the duration of a strike.
Velocity Curvature	A measure of how much the velocity changes over space, used in damping models.
Helmholtz Resonator	A model for simulating low-frequency air resonance based on the behavior of enclosed air cavity, used to model shell-air interaction.
Excitation envelope	A temporal function that modulates the amplitude of the excitation over time, shaping the strike's energy input.
Gaussian Convolution Kernel	A smooth, bell-shaped filter to extract low-frequency components. In this thesis, it is applied to velocity to separate low- and high-frequency content for frequency-dependent damping.

1 Introduction

Percussion instruments represent challenging sounds to synthesize convincingly due to their complex acoustic behaviors, rich modal structures, and highly dynamic response to playing techniques. Creating sound effects and music notes in the digital domain utilizes different mathematical and signal processing approaches. These approaches stems back to the beginning of the 1950's but witnessed major development in the 1980's, embodied in abstract synthesis techniques like wave-tables, sinusoids, and FM oscillators. The word abstract is used here because the techniques mentioned are not associated with the physical interpretation of the instruments. This means that the resulting sounds are obtained due to perceptual and mathematical principles rather than physical ones producing synthetic sounds. Later in the evolution of sound synthesis, new methods based on the processing of sampled sounds became a staple in music production, having the ability to accurately play back the acoustic quality of the sound but lacking parametric control over the sound [2].

Physical modeling represents a fundamentally different approach to sound synthesis by starting with the mathematical description of an instrument's physical behavior rather than directly manipulating sound parameters. Instead of building sounds through abstract signal processing operations, physical modeling uses differential equations to describe the displacement of a membrane, plate, or string and the propagation of the resulting sound waves. This approach inherently captures the complex interactions, nonlinearities, and dynamic behaviors that give acoustic instruments their distinctive character and expressiveness. In addition, it also becomes possible with physical modeling to generate sounds of instruments experimenting with unrealistic physical parameters[2] [3].

This thesis work have been done in collaboration with Elektron. A company known for its unique approach to electronic instruments and music design, mixing different synthesis methods to present a wide range of sound creation tools. The sound generation techniques applied in their products include subtractive synthesis, frequency modulation, wave-table synthesis, and sample-based playback. However, using physical modeling methods has not been applied yet, and there is an interest in exploring this field regarding the possibilities of creating sounds using these methods. This study is dedicated to examining and investigating how to model the sound of drums using the mass-spring system as a physical modeling method, also implementing the generated sound in real-time.

1.1 Research Questions

This research seeks to understand and implement physical modeling techniques for percussion instruments, particularly focusing on modern drums. This includes rack toms, floor toms, kick and snare drum as part of a rock and pop drum kit. The investigation progresses from theoretical comparison to practical implementation, addressing the following agenda:

1.1.1 Aim

The aim of this work is to investigate as well as compare between different physical modeling methods for realistic and expressive drum sound synthesis. The project will focus on developing a model that has a physical basis and is suitable for real-time implementation.

To specify the aim of the project in greater detail the key research questions will be defined by:

- How can different physical modeling methods be implemented to model the sound of a percussion instrument, and which one align best with our expectations?
- What are the critical parameters that can be used to simulate the modeled sound, and how can they be designed?
- Is it possible to combine different modeling techniques to achieve better results in simulating the sound desired?
- How does the implemented model work in a real-time environment, and how can it be controlled in an interactive musical setting?

Other considerations are:

- Nonlinearity involvement represented by dynamic tension and dependent damping, and its role in improving realism and natural decay mechanism.
- The influence of application of air coupling, rim reflections, and other physical aspects. Also, how it affects the drum both in performance and realism.
- The real-time performance, and weather it maintains sufficient physical accuracy of the drum sound if simplified.

1.1.2 Limitations

For practicality reasons as well as maintaining focus, some features regarding percussion modeling are not specified in detail in this project.

The thesis does not simulate room acoustics or microphone capturing characteristics since the work will focus on the direct physical behavior of the drum. Acoustic environment has a significant roll affecting the spacial impressions due to early reflections, reverberation, and room modes. Microphones also affect how the drum would sound due to the positioning, polar patterns and frequency response.

Additionally, environmental factors such as temperature and humidity are not investigated. These factors can influence the behavior of real drum sounds. Temperature affects the air density, speed of sound and membrane compliance that would shift the resonance frequency in small amounts. Humidity impacts the tension and stiffness resulting a slight de-tuning or damping variations.

Finally, playing mechanism and striking dynamics are not explored in a broader sense. Those factors would affect several aspects such as contact duration and energy transfer. In this

paper, excitation is introduced as a predefined displacement or external force applied at a specific location on the membrane.

It is also worth to mention that the approaches for modeling certain aspects such as the shell of the drums and the air coupling between the batter and resonator head are simplified. Studying and implementing these effects is rather complex. In other words, their influence applied to the model that is built in this work was insignificant compared to other mechanisms incorporated in the model.

2 Background To Modeling Techniques

To be able to model a drum, four main physical modeling methods have been chosen for their ability to be applied in real-time, a key requirement of this thesis. Each method will be described in general to illustrate how it facilitates membrane vibrations.

2.1 Finite Difference Time Domain

The Finite Difference Time Domain (FDTD) method is a numerical technique used for solving wave equations by discretizing both space and time. This technique is highly suitable in representing arbitrary shapes and is highly accurate. [4]

2.1.1 Mathematical Formulation

The Finite-Difference Time-Domain method transforms the continuous wave equation into a discrete computational problem. For a 2D membrane, this foundational equation:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \quad or \quad \frac{\partial^2 u}{\partial t^2} = c^2 \nabla^2 u \tag{1}$$

where:

$u(x, y, t)$ represents the displacement of the membrane at time t.

c is the wave propagation speed in the medium.

2.1.2 Discretization Using FDTD

The essence of the FDTD method lies in converting continuous derivatives into discrete approximations on a structured grid. Following the approach outlined by M. Kahrs and K. Brandenburg in Applications of Digital Signal Processing to Audio and Acoustics [5], represents the temporal and spatial second derivatives using central difference schemes:

$$\frac{\partial^2 u}{\partial t^2} \approx \frac{u_{i,j}^{n+1} - 2u_{i,j}^{n} + u_{i,j}^{n-1}}{\Delta t^2} \tag{2}$$

For spatial derivatives, similar approximations apply at grid points (i, j):

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}}{\Delta x^2} \tag{3}$$

$$\frac{\partial^2 u}{\partial y^2} \approx \frac{u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}}{\Delta y^2} \tag{4}$$

These discrete operators transform our wave equation into an explicit time-stepping formula:

$$u_{i,j}^{n+1} = 2u_{i,j}^{n} - u_{i,j}^{n-1} + (c\Delta t)^2 \left(\frac{u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}}{\Delta x^2} + \frac{u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}}{\Delta y^2} \right) \tag{5}$$

This update equation directly computes future values ($u_{i,j}^{n+1}$) from current and past states, creating a leapfrog scheme characteristic of FDTD methods.

2.1.3 Boundary Conditions

The FDTD model requires appropriate boundary conditions on the 2D grid where each node stores displacement values $u_{i,j}^{n}$ at discrete time steps. For an accurate representation of a physical drum membrane, as described by [6], the simulation implements two critical boundary types:

- **Fixed boundaries:** For a drum, the edges are rigidly clamped, giving $u = 0$ at all boundary nodes.
- **Damping mechanisms:** To account for energy dissipation in real drums, the model should incorporate absorbing boundary conditions that progressively attenuate outgoing waves. This can be achieved through a damping coefficient α applied near the boundaries:

  $$u_{i,j}^{n+1} = (1 - \alpha_{i,j}) u_{i,j}^{n+1} \tag{6}$$

  where $\alpha_{i,j}$ increases gradually toward the boundary edges.

These boundary conditions ensure that the simulation captures both the geometric constraints and the energy loss characteristics of the physical drum membranes.

2.2 Delay Line

The 1D digital waveguide provides the foundation for delay line modeling of wave propagation. It is based on the traveling wave solution to the 1D wave equation by Smith and Van Duyne in Physical modeling with the 2-d digital waveguide mesh [7]:

$$u_{tt}(t, x) = c^2 u_{xx}(t, x) \tag{7}$$

The general solution consists of two traveling wave components:

$$u(t, x) = g^{+}(x - ct) + g^{-}(x + ct) \tag{8}$$

where $g^{+}(x - ct)$ represents a wave traveling to the right and $g^{-}(x + ct)$ represents a wave traveling to the left, both at speed c. In the digital domain, this is implemented using two bidirectional delay lines that store and circulate the traveling wave components in opposite directions. Each delay line maintains a series of sample values representing the wave at discrete spatial positions, with samples shifting through the delay elements to simulate wave propagation. The right-traveling wave $g^{+}(x - ct)$ is stored in one delay line where samples

advance from left to right with each time step, while the left-traveling wave $g^-(x+ct)$ is stored in another delay line where samples advance from right to left, effectively creating a discrete-time analog of the continuous wave equation. For discrete time and space with sampling interval T and spatial step X, the Courant-Friedrichs-Lewy stability condition requires:

$$cT/X \leq 1 \tag{9}$$

This ensures that the wave cannot travel more than one spatial step per time step, maintaining numerical stability by preventing information from propagating faster than the discretization can handle.

The 2D digital waveguide mesh is an extension of the 1D digital waveguide used for simulating 2D wave propagation. It is constructed out of bidirectional delay units and scattering junctions that carry traveling wave components between discrete points in a spatial grid.

2.2.1 Mathematical Formulation

In a 2D waveguide mesh, the membrane is discretized into a network of nodes, each connected to its neighboring nodes through bidirectional delay lines. The traveling waves propagate through these connections, simulating the spread of energy over time. The fundamental update equation is derived from the scattering junction equations at each node further described by Smith and Van Duyne [7]:

$$v_{J_{l,m}}(n) = \frac{v^+_{1_{l,m}}(n) + v^+_{2_{l,m}}(n) + v^+_{3_{l,m}}(n) + v^+_{4_{l,m}}(n)}{2} \tag{10}$$

$$v^-_{i_{l,m}}(n) = v_{J_{l,m}}(n) - v^+_{i_{l,m}}(n) \tag{11}$$

where $v_{J_{l,m}}(n)$ represents the junction velocity at position (l,m) and time n, and $v^+_{i_{l,m}}(n)$ and $v^-_{i_{l,m}}(n)$ represent the incoming and outgoing wave components at the four ports of each junction.

Through algebraic manipulation of these scattering relations and the interconnection constraints, this reduces to the standard finite difference approximation of the 2D wave equation:

$$\begin{aligned} v_{J_{l,m}}(n) - 2v_{J_{l,m}}(n-1) + v_{J_{l,m}}(n-2) = \frac{1}{2}[v_{J_{l,m+1}}(n-1) - 2v_{J_{l,m}}(n-1) + v_{J_{l,m-1}}(n-1)] \\ + \frac{1}{2}[v_{J_{l+1,m}}(n-1) - 2v_{J_{l,m}}(n-1) + v_{J_{l-1,m}}(n-1)] \end{aligned} \tag{12}$$

This update rule ensures that waves propagate accurately while maintaining numerical stability and energy conservation.

2.2.2 Wave Components and Scattering

The traveling wave components in each direction are governed by the bidirectional delay lines, where:

- v^+ represents incoming wave components at each junction port
- v^- represents outgoing wave components at each junction port
- Each unit delay represents wave propagation between adjacent nodes

The scattering junctions at each node distribute incoming waves to the connected delay lines according to the physical constraints that: (1) velocities of all connected elements at the junction must be equal, and (2) forces at each junction must sum to zero. This ensures proper energy distribution and conservation throughout the mesh.

The mesh implements a wave propagation speed of $c = 2^{-1/2} \approx 0.707$ between junction positions. This value emerges from the 2D mesh geometry where waves can propagate both horizontally/vertically (distance 1) and diagonally (distance $\sqrt{2}$). To satisfy the CFL stability condition from equation 9 for unit time steps and unit spatial grid spacing, and to ensure isotropic wave propagation in all directions, the wave speed is set to $c = 1/\sqrt{2}$, providing optimal stability while maintaining directional consistency throughout the mesh structure.

2.2.3 Boundary Conditions

Like finite difference time domain (FDTD) methods, the waveguide mesh requires boundary conditions to handle reflections and energy dissipation:

- **Clamped boundaries**: Nodes at the boundaries reflect incoming waves with inversion, simulating fixed drum edges
- **Free boundaries**: Waves reflect without inversion for unclamped plate models
- **Absorbing boundaries**: Special termination conditions can be introduced to mimic realistic energy dissipation

2.3 Modal Synthesis

This method is considered to be a powerful technique used to simulate physical instrument, especially percussion. The principle of this method is to model the instrument vibration by decomposing its oscillations into independent modes. For percussion, there will be a dynamic interaction between the resulting modes of the vibration and other factors such as the excitation force, damping and nonlinear effects.

The movement of the drumhead can be described with modes, which are oscillation patterns. Each mode represents a standing wave, oscillates at a certain frequency, and contributes to the overall sound of the drum. To study these patterns, The calculation of the displacement of each point of the drum head is performed using the following mathematical equation proposed by S. Bilbao [2]:

$$u(r,\theta,t) = \sum_{m,n} A_{mn}\phi_{mn}(r,\theta)e^{-\sigma_{mn}t}\cos(\omega_{mn}t + \phi_{mn}) \tag{13}$$

where:

$A_{m,n}$ is the modal amplitude.
$\phi(r,\theta)$ is the mode shape function.
σ_{mn} is damping coefficient.
ω_{mn} is the modal frequency.
ϕ_{mn} is the phase offset.

The mode shape function $\phi(r,\theta)$ describes the spatial distribution of the displacement of each vibration mode in the drum head. In terms of a circular membrane, it is typically represented by Bessel functions described further in Section 2.6.2. In trigonometric terms, the Bessel function captures different regions of the drum's movement during the oscillation. Mode shapes determine where vibrations occur, with nodes and antinodes, and are calculated in general using the following equation proposed by N. H. Fletcher and T. D. Rossing [6]:

$$\phi_{mn}(r,\theta) = J_m\left(\frac{\alpha_{mn}r}{R}\right)\cos(m\theta) \tag{14}$$

where:

J_m is the Bessel function of the first kind
α_{mn} is the eigenvalues so that $J_m(\alpha_n) = 0$
R is the drumhead radius.

α_{mn} are the roots of the Bessel function which define the resonant frequency of the drum membrane. When $J_m(x) = 0$, no radial displacement happen at the fixed boundary of the drumhead.

The modal frequencies ω_{mn} are calculated using the following equation:

$$\omega_{mn} = c\left(\frac{\alpha_{mn}}{R}\right) \tag{15}$$

c is the speed of wave in the membrane in and is calculated using the formula:

$$c = \sqrt{\frac{T}{\rho}} \tag{16}$$

where T is the tension of the membrane in N/m and represents the force per unit length applied to the drumhead. ρ is the mass per unit area. Higher tension increases the wave speed, while higher mass slows it down. The concept of tension is further described in Section 2.6.1.

In terms of the damping in the modal synthesis method, Each mode decays over time due to energy loss:

$$\sigma_{mn} = \frac{\omega_{mn}}{Q} \tag{17}$$

where Q is the quality factor that controls the damping, meaning how long a mode vibrates before fading. Additionally, damping is considered to be frequency dependent, which means:

$$\sigma_{mn} = \sigma_0 + \alpha\omega_{mn} \tag{18}$$

where σ_0 is baseline damping, and α scales high frequency damping.

2.4 Mass Spring

This method simulates the dynamic behavior of vibrating systems, like a drum membrane, by discretizing it into a network of masses and springs. This approach captures the essential physical properties of the system, including mass distribution, tension, and damping. A drum membrane can be modeled to be a grid with two dimensions. This grid consists of small masses where each represents a small portion of the membrane. The springs represent the tension forces between neighboring masses. The discretization of this system includes a transformation into a set of differential equations that can be solved numerically.

2.4.1 Fundamental Equations

Newton's second law determines the motion of each mass, and given as the following:

$$m\left(\frac{d^2u}{dt^2}\right) = \mathrm{F}_{\mathrm{Total}} \tag{19}$$

$$\mathrm{F}_{\mathrm{Total}} = \mathrm{F}_{\mathrm{damping}} + \mathrm{F}_{\mathrm{spring}} \tag{20}$$

where m is the mass of the node in kg, u is the displacement vector of the mass in meters, and $\mathrm{F}_{\mathrm{Total}}$ is the total force acting on the mass in Newtons.

Hooke's law describes the force applied by a spring and it is proportional to its extension or compression:

$$\mathrm{F}_{\mathrm{spring}} = -k(u - u_0) \tag{21}$$

where k is the spring stiffness constant in N/m, u is the displacement vector, and u_0 is the rest position vector m.

The energy loss due to friction and air resistance is represented by the damping forces, and expressed as following:

$$\mathrm{F}_{\mathrm{damping}} = -b\mathrm{v} \tag{22}$$

where b is the damping coefficient that can be determined experimentally based on real drum behavior or estimated using material properties such as Young's modulus and internal friction. v is the velocity of the mass [8].

2.4.2 Boundary Conditions

Boundary condition affect wave reflections, energy dissipation, and overall system stability. At the fixed edges of the membrane, $u = 0$ at the boundary. This condition prevents motion at the edges, causing wave reflection with an inverted phase.

Other boundary conditions are the damped boundaries implemented by applying a damping force proportional to velocity at the boundaries, and periodic boundaries that simulates a system where the edges wrap around [2].

2.5 Summary of the Techniques

One of the goals was to investigate each method and evaluate its strengths and weaknesses. The approach for the theory implementation was to build a simplified model in MATLAB that would simulate a drum sound with realistic parameters gathered from the book The Physics of Musical Instruments by Rossing and Fletcher [6], such as membrane thickness, density, Young's modulus, etc. These parameters were used as a baseline for every method implementation to set a standard model, which would help evaluate the models better.

2.5.1 Finite Difference Time domain

The Finite Difference Time Domain (FDTD) method has its ability to simulate complex wave interactions across a drumhead with a high degree of accuracy and adapts well to a system with arbitrary shapes. Compared to the other methods, FDTD allows for high accuracy for the simulation of wave propagation, as well as reflection in both time and space. This method has proven to generate sounds that closely mimic the characteristics of a drum sound.

Despite its advantages, the FDTD method comes with a substantial computational cost, which could potentially be an obstacle when trying to incorporate it in less powerful machines. It requires significant processing power to calculate wave propagation at each discrete point across the drum's surface for each time step, leading to a limitation since real-time sound generation is desired. Depending on the processing power of the machine, it is possible to reduce the mesh grid down to fewer points to be able to run it, but with the drawback of losing fidelity of the sound.

Furthermore, the FDTD method can have stability issues, where the latter is important due to the relationship between the time step size, spatial discretization, and the speed of sound through the medium—closely tied to the Courant-Friedrichs-Lewy (CFL) condition [2][9]. If this condition is not managed properly, it can lead to numerical instability, where small errors in calculation rapidly grow and degrade the simulation's accuracy over time or start to self-oscillate.

Also, FDTD is found to be limited regarding incorporating nonlinear interactions, which has its significance in musical acoustics. While linear approximations are often sufficient for many scenarios, real-world musical sounds, especially those produced by drums, involve complex nonlinear behaviors such as variable tension influenced by strike velocity and frequency, in addition to other experimental setups that are sought after in the work collaboration with Elektron.

In summary, while the FDTD method has been embraced for its ability to produce good drum sounds, its practical application is too limited by high computational requirements and stability concerns. Furthermore, the method has an inherent linear nature, which means that more adaptations are necessary to fully capture the desired non-linearities of a drum sounds.

2.5.2 Delay Line

The delay line system is a well documented method that has been used in many products before, but mainly for one-dimensional models. There are cases where a delay line system have been used in two dimensions, and it is capable to simulate wave propagation. However, the challenge comes when dealing with reflective edges in a two-dimensional surface, especially circular membranes, and producing wanted drum sounds.

As a branch to delay line systems, banded waveguides have been investigated which offers a method for capturing the nuances of multiple resonant modes. Much like modal synthesis this technique was proven to be too linear for replicate the complex behavior of a drum.

Although delay lines have an important role in physical modeling with potential applications in simulating certain aspects of drum sounds, they were less intuitive as a drum model and not as good sounding as other methods.

2.5.3 Modal Synthesis

This approach offers a good demonstration of the object properties. For a percussion instrument, a structured approach could be followed, which includes defining the modal frequencies based on the physical parameters, implementing an excitation to trigger the response, damping to create energy decays.

Compared to other methods such as FDTD or mass-spring networks, modal synthesis offers high computational efficiency which reduces the cost of solving differential equations, using the summation of precomputed modes. It also provides clear relationship between the physical parameters and the output sound. In addition, increasing the resonant modes and damping models can be done without significant computation complexity.

However, applying nonlinear effects such as tension variations and damping requires additional modeling complexity, which makes this model not particularly ideal for handling nonlinearity. Also, the mapping of expressive playing techniques like dynamic strokes regarding percussion, remains a challenge since it requires finding external approaches into excitation modeling. Such challenges have led to not using this method as it will be a liability regarding desired characteristics of the generated sound.

2.5.4 Mass Spring

The implementation of this method includes capturing the wave propagation and introducing nonlinear interactions such as tension and damping, as well as realistic vibrational modes.

The model captures natural wave propagation and allows tension modulation increasing the pitch with higher amplitude strikes as a result. It is flexible and can be extended to include shell coupling, air interaction, and different excitation behaviors, giving the advantage of simulating different types of drumheads. When it comes to real-time implementation, it can be optimized for running interactive drum synthesis.

However, a high-resolution network can make the computation of the model expensive since it requires many calculations per time step. The method also come with some challenges due to implementing nonlinearity as it requires tuning the parameters carefully. In addition, the boundary handling can become complicated when dealing with coupling mechanisms.

2.5.5 Summary

In Table 1 the different methods are summarized to their advantages, limitations and integration into a real time situation.

Table 1: Comparison of Physical Modeling Methods

Method	Advantages	Limitations	Implementation & Real-Time
FDTD	Accurate wave interactions; produces authentic drum sounds	High computational cost; stability issues (CFL condition); limited nonlinearity	High processing needs; grid reduction lowers fidelity; limited by computation and stability demands
Delay Line	Well-documented; Fast calculations while simulating wave propagation	Problems with recreating the sound of circular membranes	Efficient for real time implementation but lacks the realism wanted
Modal Synthesis	Efficient computation; clear parameter relationships.	Poor nonlinearity handling; difficult mapping of playing techniques	Complex implementation of nonlinear effects; scalable mode complexity; liability for realistic real-time sound production
Mass-Spring	Captures 2D behavior; dynamic parameter response; natural wave propagation.	Computationally expensive; requires careful parameter tuning.	Optimizable for real-time implementation; dynamic capability for development.

To evaluate the comparative performance and sound quality of these approaches, all methods were implemented as modular MATLAB prototypes. This allowed for systematic assessment of computational efficiency, sound fidelity, and implementation complexity under controlled conditions.

Based on this comparative analysis, the mass-spring system demonstrates significant advantages for development and real-time implementation compared to other methods. It effectively captures 2D membrane behavior and nonlinear effects while accommodating sophisticated excitation force models, resulting in enhanced realism. The system's ability to dynamically respond to changes in force, tension, and other physical parameters, combined with its inherent flexibility for optimization, makes it particularly suitable for real-time applications. In contrast, approaches such as digital waveguides show limitations in 2D scenarios and reduced flexibility for parameter manipulation and model extension.

2.6 Complimentary Studies

Considering drum modeling, there are some elements that need to be explained, since they represent key factors for the sound characteristics of a drum. However, they will be described concisely since they will be considered more related to parametric design and realism, not the general implementation of physical modeling.

2.6.1 Tension

One of the most important mechanisms applied to drums is the tension, as it has a significant role regarding the pitch of the sound. It has also a direct relationship with the wave speed in the membrane.

A simplified solution to *Newton's Second Law* and the equation of motion can be described as:

$$F = m \cdot a \rightarrow F_{Total} = T \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) dxdy \tag{23}$$

and:

$$F_{Total} = \rho dxdy \frac{\partial^2 u}{\partial t^2} \tag{24}$$

where F is the force acting in the edges of a membrane from the tension T which are dependent on the vertical displacement as u. Solving for the wave speed c can be found by considering the mass per unit area ρ from equation 16.

2.6.2 Modes

The vibrational behavior of a membrane, such as a drumhead, can be described using Bessel function. This function helps characterize how the membrane moves in circular patterns and determine its natural vibration modes. The simplest vibration mode involves the entire membrane moving up and down together, while higher modes introduce more complex patterns with multiple nodal regions (displacement = 0).

Bessel functions arise naturally in problems with cylindrical symmetry, such as heat conduction in a cylinder, electromagnetic waves in a circular waveguide, and vibration modes of circular membranes. In particular, the study of vibrating circular membranes requires solutions to the wave equation in cylindrical coordinates, leading to Bessel functions of the first kind, denoted as $J_n(x)$ [6].

Bessel functions are solutions to Bessel's differential equation:

$$x^2 \frac{d^2 y}{dx^2} + x \frac{dy}{dx} + (x^2 - n^2) y = 0, \tag{25}$$

where n is a real or complex number known as the order of the function. The equations referenced in this section are taken from Chapter 3 of The Physics of Musical Instruments by Neville H. Fletcher and Thomas D. Rossing [6].

The vibration modes of a circular membrane are described by the two-dimensional wave equation in polar coordinates:

$$\nabla^2 u - \frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} = 0, \tag{26}$$

where $u(r, \theta, t)$ is the displacement of the membrane at radial distance r, angular coordinate θ, and time t. Separation of variables leads to a radial equation involving Bessel functions:

$$r^2\frac{d^2R}{dr^2} + r\frac{dR}{dr} + (k^2r^2 - n^2)R = 0, \tag{27}$$

where k is the wave number and radius $R(r)$ must satisfy boundary conditions at the membrane's edge. Leading to the solution found in Equation 14 solving the mode shape of a membrane.

The characteristic modes are given by the zeros of the Bessel function $J_n(x)$, meaning that the natural frequencies of the membrane are determined by the roots $j_{n,m}$ of $J_n(x)$. The fundamental vibration mode corresponds to the first root of $J_0(x)$, which defines the lowest frequency of vibration as illustrated in figure 1 and their roots in table 2.

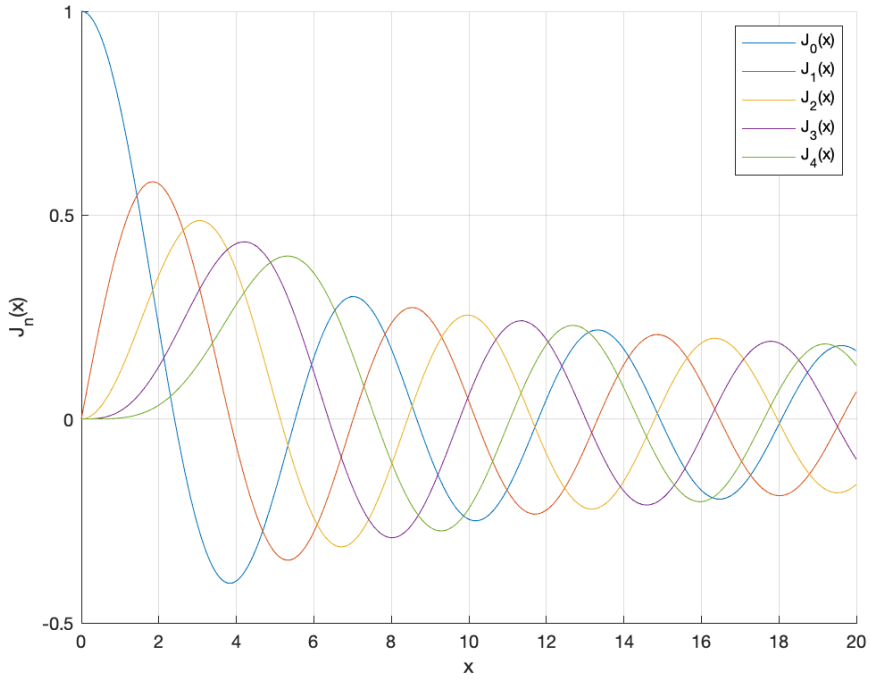


Figure 1: Bessel functions of the first kind $J_n(x)$ and their roots, which determine the theoretical modal frequencies of a circular drum membrane.

Table 2: Bessel roots

J_0	J_1	J_2	J_3	J_4
2.4048	3.8317	5.1356	6.3802	7.5883
5.5201	7.0156	8.4172	9.7610	11.0647
8.6537	10.1735	11.6198	13.0152	14.3725
11.7915	13.3237	14.7960	16.2235	17.6160
14.9309	16.4706	17.9598	19.4094	-
18.0711	19.6159	-	-	-

These vibration modes are relevant for understanding how drums produce sound and why different striking positions lead to different tones. The Bessel functions provide a powerful mathematical framework to describe these modes and predict their frequencies. The first few roots of $J_0(x)$ dictate the dominant resonant modes, which can be particularly valuable when designing realistic drum synthesis, independent of the physical modeling approach used.

When implementing physical models such as the mass-spring systems, it's important to note that these Bessel modes won't inherently appear by default. Instead, the Bessel function solutions serve as an analytical reference against which numerical models can be validated. By comparing the emergent modes and frequencies from the physical model to the theoretically predicted Bessel modes, one can quantitatively assess how accurately the implementation captures the true physical behavior of the membrane. This validation approach allows for systematic refinement of the model parameters (such as mass distribution, spring constants, and mesh density) until the simulated vibration patterns converge to match the expected modes described by the Bessel functions or closely relate to them.

3 Method

The method is structured in three main stages. The first introduces the basic model with mass-spring system, putting the foundation and implementing only the most simple physical principles of membrane dynamics, such as mass and stiffness. The second stage focuses on the development and refinement of the offline model. This includes incorporating additional parameters that contribute to more sophisticated mechanism to represent nonlinearity in the model, leading to capturing more nuances of modal behavior, and energy distribution manipulation. Finally, real-time implementation is described, showing how the mass-spring system can be adapted for real-time performance. New constraints are introduced and represented by computational efficiency and numerical stability.

3.1 Basic Model

The basic model stems from the comparison between the different method where the fundamental principles of Section 2.4 was used. The representation of the drum only used a handful of masses connected by springs, rather than implementing a full membrane with many grid points.

This elementary model approached the membrane as a simple mechanical system where a small number of mass points were arranged in a basic pattern and connected by springs. The central mass point represented the primary resonator, with surrounding points providing basic modal behavior. This simplified system moved more like a piston than a proper membrane, but established the core physical relationships between mass, tension, and resulting oscillation.

By starting with this reduced system, it was possible to verify basic behaviors such as:

- Energy conservation in the absence of damping.
- Natural frequency scaling with tension and mass.
- Basic principles of excitation response.

This approach allowed to test parameter relationships before introducing the computational complexity of full grid system. Once these fundamentals were established, more advanced procedure was progressed including grid structures—both Cartesian and polar coordinate systems—which will be described in detail in the offline model development Section.

3.2 Offline Model

This section concentrates on the development of the physical model of the drum using a mass-spring system. It expands to introduce the elements that enhance accuracy and dynamic behavior including grid discretization methods, mass distribution approach, frequency- and amplitude-dependent damping, shaped excitation forces, and nonlinear tension modeling.

3.2.1 Grid Discretization

Grid discretization is an important step to represent the circular membrane of the drum. For the development of the drum simulation model, two types of grid discretizations have been used. While developing the offline model that focuses on realism and do not have any restriction of computation time a Cartesian grid have been implemented for its mass distribution and tension uniformness (See Figure 2). The real-time model, however, needs much faster calculations and it was found that a polar grid setup was beneficial with fewer grid points for the perceptual likeness to a drum. Both concepts, however, will be described in this section.

The offline model is specified to have a circular shape for representing the drum membrane as described by Morgan [3]. The membrane in this approach is sampled into discrete points, which are masses. The continuous mechanical interactions are approximated by spring connections between the neighboring points as described by Bilbao [2]. In other words, each point becomes a small patch of the drumhead, and each connection represents tension-induced restoring forces along the membrane. Mass and spring parameters are derived from realistic dimensions and material properties of the drum.

To create a 2D Cartesian grid, the points are distributed in normalized coordinates over [-1,1] in both x and y directions where:

$$X(i,j), Y(i,j) \in [-1,1], i,j = 1, ..., \text{grid size} \tag{28}$$

Implementing the Cartesian coordinate system will create a rectangular grid. In order to reshape the grid to match the geometry of the drumhead, the circular domain is defined according to the following:

$$\text{circular boundary}(i,j) = \begin{cases} 1, & \text{if} \quad X(i,j)^2 + Y(i,j)^2 \leq 1 \\ 0, & \text{otherwise} \end{cases} \tag{29}$$

For the rim reflection simulation of the membrane, a boundary tolerance is defined, and then a circular mask is determined due to the following:

$$\text{boundary\_mask}(i,j) = \begin{cases} 1, & \text{if} \quad \sqrt{X(i,j)^2 + Y(i,j)^2} \geq 1 - \frac{2}{\text{grid\_size}} \\ 0, & \text{otherwise} \end{cases} \tag{30}$$

Finally, rim marking is used to detect the points where the circular drum membrane ends. This helps apply reflections and boundary conditions only at those points.

With the procedure above, The inner area of the created grid would behave like a free vibrating membrane, where the boundary (rim) enforces reflection and energy loss using reflection coefficients. The Cartesian coordinate system used in this model allows fast and simple computation while the created mask maintains the circular behavior.

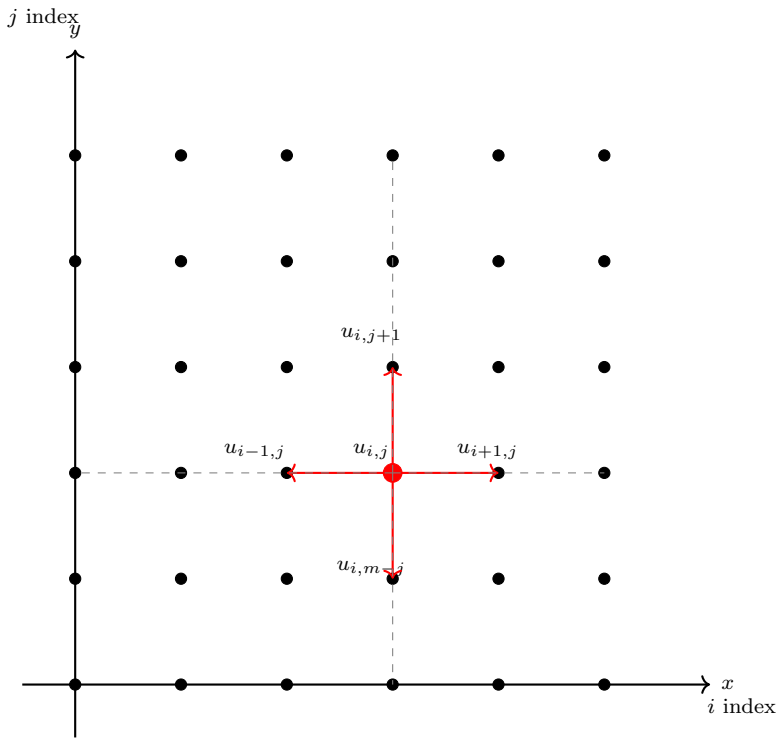


Figure 2: Discrete 2D Cartesian grid used in the simulation. The central node $u_{i,j}$ is connected to its four neighbors, forming the 5-point stencil used in finite difference and mass-spring methods.

During the development phase the investigation of polar coordinates was condoned. This grid setup was then used for the real-time implementation. This approach naturally accommodates the circular geometry of the drum, eliminating the need for boundary masking as required in the Cartesian system. The polar grid discretization provides a more direct representation of Bessel root solution that characterize drum vibrations even with fewer grid points.

The polar coordinate system discretizes the membrane using radial and angular coordinates, where each point is defined by its distance from the center (r) and the angle (θ) it makes with the positive x-axis. To create a 2D polar grid, the points are distributed in normalized coordinates with:

$$\begin{aligned} r(i) &\in [0,1], \quad i = 1, ..., N_r \\ \theta(j) &\in [0, 2\pi], \quad j = 1, ..., N_\theta \end{aligned} \tag{31}$$

where N_r is the number of radial divisions and N_θ is the number of angular divisions. The Cartesian equivalents can be computed as:

$$\begin{aligned} X(i,j) &= r(i)\cos(\theta(j)) \\ Y(i,j) &= r(i)\sin(\theta(j)) \end{aligned} \tag{32}$$

Special consideration must be given to the center point ($r = 0$), where the angular coordinate becomes undefined. This singularity is handled by placing a single mass at the center connected to all masses in the first radial ring.

The spacing between grid points varies with the radial distance in a polar grid. Points are more densely packed near the center and become increasingly separated toward the boundary. This non-uniform spacing needs to be accounted for when calculating the spring constants between neighboring mass points. The spring constants are adjusted proportionally to the local arc lengths and radial spacing according to:

$$\begin{aligned} k_{radial}(i) &\propto \tfrac{T}{r(i+1)-r(i)} \\ k_{angular}(i,j) &\propto \tfrac{T}{r(i)\cdot(\theta(j+1)-\theta(j))} \end{aligned} \tag{33}$$

where T represents the membrane tension. This adjustment ensures that the physical properties of the membrane are correctly preserved throughout the simulation, based around Hooke's law 21.

For the rim reflection in polar coordinates, the boundary condition is simply applied at the outermost ring of mass points where $r = 1$. The reflection coefficient can be directly applied to these points without requiring a boundary mask:

$$\text{boundary\_points}(i,j) = \begin{cases} 1, & \text{if} \quad r(i) = 1 \\ 0, & \text{otherwise} \end{cases} \tag{34}$$

The polar coordinate system offers computational advantages for certain algorithms, especially those involving modal analysis or radial wave propagation despite having few grid points and relating very close to the solution to the Bessel function [6].

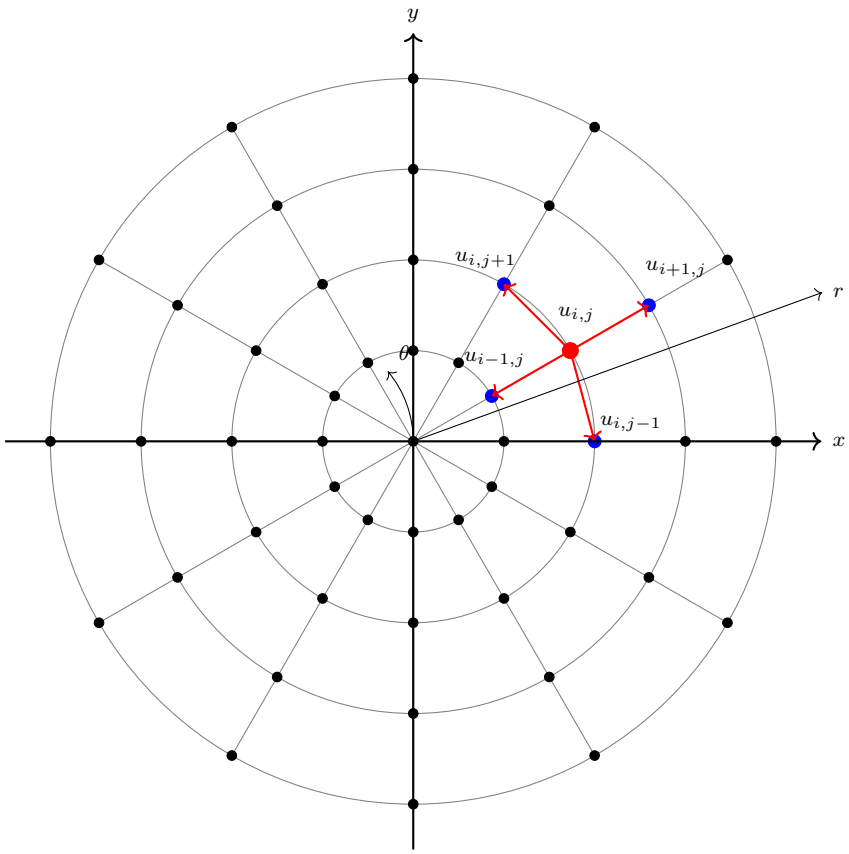


Figure 3: Discrete 2D polar grid used in the real-time implementation. The central node $u_{i,j}$ (in red) is connected to its four neighbors in the radial and angular directions, with spacing that varies based on radius.

3.2.2 Mass Distribution

After discretizing the drum membrane into a grid of points, a nonuniform mass distribution is introduced to simulate material construction irregularities and performance subtleties of drumheads. This variation contributes to more complex behavior and rich sound textures [10]. To obtain these variations of the masses, radial weighting and controlled Gaussian noise are utilized.

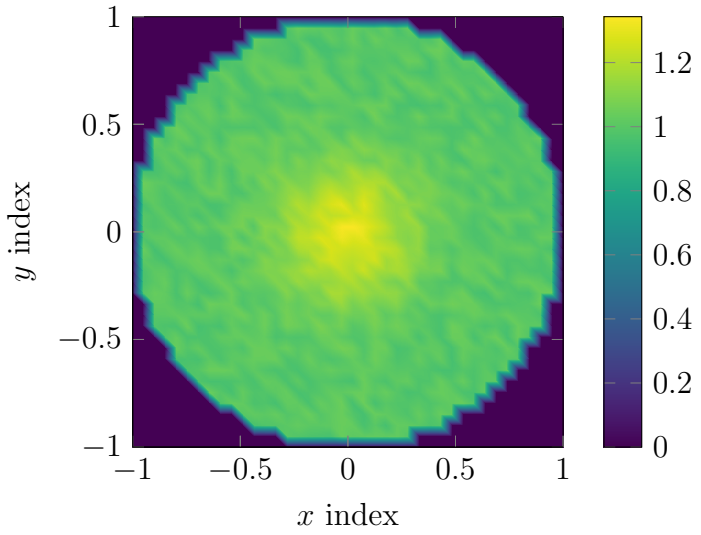


Figure 4: Mass distribution with radial and random variations over a circular membrane.

Figure 4 above illustrates the mass distribution across the membrane. The colors show relative mass scaling normalized to 1.

Each point in the grid is assigned a radial coordinate (r_{ij}) based on its Euclidean distance from the center. A radial weighting is applied to modulate the mass distribution from center to rim. The variation is exponential and controlled by a specified factor (γ):

$$m_{\text{radial}}(i,j) = \exp(\gamma \cdot (r_{ij} - 1)) \tag{35}$$

if $\gamma > 0$, mass increases toward the rim, otherwise it becomes concentrated at the center as it is shown in figure 4 above. This radial pattern introduces modal de-tuning effects, and influences the modal spacing and decay behavior [2].

To simulate the material texture, a small Gaussian noise term is added to each node using the following equation:

$$m_{\text{noise}}(i,j) = 1 + \epsilon \cdot N(0,1) \tag{36}$$

where ϵ is the noise strength and $N(0,1)$ is a sample from a standard normal distribution. Finally, each mass is calculated as follows:

$$m_{ij} = \rho \cdot h \cdot .A_{\text{node}} \cdot m_{\text{radial}}(i,j) \cdot m_{\text{noise}}(i,j) \tag{37}$$

where ρ is the membrane's material density in kg/m^2 , h is the thickness in m, and A_{node} is the physical area per grid point (Δx^2). To avoid instabilities in under-weighted regions, a minimum mass threshold is applied at each mass node.

3.2.3 Damping

The decay behavior of the drumhead is implementing a dynamic damping mechanism that combines three ingredients to enhance the realistic modeling of the sound, which are:

- Frequency-dependent damping separating low and high-frequency velocity components.
- Amplitude-dependent nonlinear damping that is based on displacement magnitude.
- Curvature based damping which is related to the membrane's velocity Laplacian.

Looking into the physical behavior of a drum membrane, this damping approach contributes to high-frequency modes decaying faster than low modes, and energy dissipation increases with membrane curvature and displacement amplitude [6]. The total damping force modeled will act opposing the velocity at each point in the circular grid. This force will reduce the kinetic energy over time and contribute to the evolution of the sound.

To apply frequency-dependent damping, the velocity at each grid point will be separated into low- and high-frequency using a 2D Gaussian convolution kernel. This process consists of sliding a small kernel matrix across the velocity grid and at each location. By applying this approach the weighted sum of neighboring values is calculated. The following formula is the 2D convolution of the velocity matrix $v(i, j)$ with a Gaussian kernel $G(i, j)$ [11].

$$v_{\text{low}}(i, j) = (v * G)(i, j) = \sum_{m=-M}^{M} \sum_{n=-N}^{N} v(i - m, j - n)\, G(m, n) \tag{38}$$

$G(m, n)$ is a discrete sampled Gaussian defined as:

$$G(m, n) = \exp\left(-\frac{m^2}{2\sigma_x^2} - \frac{n^2}{2\sigma_y^2}\right) \tag{39}$$

where σ_x, σ_y control the spread of the filter in x and y directions.

The Gaussian convolution is employed to smooth out local fluctuations in the velocity field as well as separate slow moving from fast oscillating components. In other words, low- from high frequencies. This application is important to make the damping differentiate between slow and fast motions.

After the frequency separation, The damping is applied on both low- and high frequencies according to the following:

$$d_{\text{low}}(i,j) = d_{\text{base}} \times f_{\text{low}} + d_{\text{nonlinear}} \times |v_{\text{low}}(i,j)| \tag{40}$$
$$d_{\text{high}}(i,j) = d_{\text{base}} \times f_{\text{high}} + d_{\text{nonlinear}} \times |v_{\text{high}}(i,j)| \tag{41}$$

where:
d_{base} is a constant linear damping coefficient,
f_{low}, f_{high} are low- and high-frequency damping scaling factors respectively,
$v_{\text{low}}(i,j)$, $v_{\text{high}}(i,j)$ are the low- and high-passed filtered velocity at node (i,j),
and $d_{\text{nonlinear}}$ is a coefficient controlling amplitude-dependent nonlinear damping.

Regarding the amplitude-dependent damping, the displacement damping term is introduced to the total damping mechanism to account for additional energy loss that represents the increased internal friction happening in the membrane. Displacement damping is applied at each grid node and defined as:

$$d_{\text{disp}}(i,j) = \gamma_{\text{disp}} \times \mid u(i,j) \mid \tag{42}$$

where γ_{disp} is a small positive constant that controls the strength of displacement-dependent damping. This process ensure that the damping increases proportionally as the displacement become bigger, introducing faster decay of large amplitude oscillations.

The last ingredient to be added to the total damping is the velocity Laplacian-based damping. This term is added to the overall damping model to capture the increased energy dissipation that comes with the local curvature changes [6]. At each grid node (i,j) this damping term is defined as the following:

$$d_{\text{lap}}(i,j) = \gamma_{\text{lap}} \times \left|\nabla^2 v(i,j)\right| \tag{43}$$

where γ_{lap} is a positive scalar controlling the strength of curvature damping, and $\nabla^2 v(i,j)$ is the discrete Laplacian (second spatial derivative) of the velocity:

$$\nabla^2 v(i,j) = v(i+1,j) + v(i-1,j) + v(i,j+1) + v(i,j-1) - 4v(i,j) \tag{44}$$

With this term, the high-frequency spatial structures are targeted to ensure faster decay of the sharp and rapidly changing motions.

All the damping terms above are then added to form an overall and dynamic damping model that is updated at every time step and applied as a force against the velocity during the main simulation.

3.2.4 Tension

To model the tension, an integration between linear tension and nonlinear stiffness modulation is applied. With this approach, the increasing resistance to deformation at large displacements would be captured [2] [10]. When the membrane is discretized into a grid of masses, their interaction is modeled using spring force which is calculated depending on adjacent displacement differences at each mass point. Then, the nonlinear stiffness model is applied.

Assuming that all the springs share the load equally in the linear system, a uniform tension across the membrane surface is created in the form of a linear spring constant. This constant reflects the amount of force per unit deformation when the nonlinear effects are not present. It means that the relationship between force and deformation follows Hooke's law [12]. Assuming the base tension of the membrane is evenly distributed, the spring constant (k) is then derived taking into account the number of grid points and spacing:

$$k = \frac{T}{h \cdot N} \tag{45}$$

where (T) is the base tension on the surface in N/m, (h) is the grid spacing in meter, and (N) is the number of masses along one axis. The base spring constant sets the fundamental frequency of the membrane, the base wave speed, and the reference point for nonlinear modulation which is used with larger displacements [2].

The local deformation in the membrane surface is represented by the displacement differences between the neighboring masses. These differences will be the input to the calculation of the nonlinear spring stiffness, and they capture the directional stretching and compression in the grid. The numerical approach used to compute the displacement differences is based on the first order finite differences which is as the following:

$$\begin{aligned} \Delta_x^+(i,j) &= u(i+1,j) - u(i,j) \\ \Delta_x^-(i,j) &= u(i,j) - u(i-1,j) \\ \Delta_y^+(i,j) &= u(i,j+1) - u(i,j) \\ \Delta_y^-(i,j) &= u(i,j) - u(i,j-1) \end{aligned} \tag{46}$$

where $u(i,j)$ is the displacement at node (i,j), and Δ_x^+, Δ_y^+ are forward and backward differences in the x and y directions respectively.

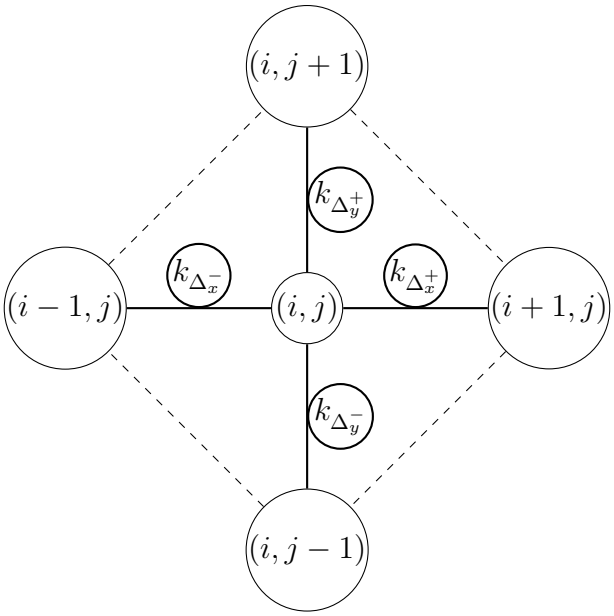


Figure 5: Schematic of a membrane node (i,j) connected to its four neighboring nodes via dynamic springs. Each spring stiffness depends on the displacement difference, modeling nonlinear tension behavior.

To model the non-linear behavior of the membrane, the increase in local stiffness is calculated based on the increase of displacement, since the tension is increasing with the stretch due to geometric and material non-linearities. Thus, the spring stiffness will be modified between each pair of points depending on the magnitude of local deformations:

$$k_\Delta(i,j) = k \cdot \left(1 + \alpha \cdot |\Delta(i,j)| + \beta \Delta(i,j)^2\right) \tag{47}$$

Equation 47 above represent the stiffness $k_\Delta(i,j)$ of each spring link connecting to neighboring masses which is computed in all directions. α is the coefficient that controls first-order stiffness increase, and the β coefficient modulates second-order nonlinear elasticity. Both terms are considered to be displacement-dependent correction terms that would capture the rise in frequency during the attack and play a significant role in de-tuning under force [2].

The next step is to calculate the spring forces. The net force acting on each node (mass) is computed as the sum of the contributions from its direct neighbors in the horizontal and vertical directions:

$$\begin{aligned} F_{x^+}(i,j) &= k_{x^+}(i,j) \cdot \Delta_x^+(i,j) \\ F_{x^-}(i,j) &= k_{x^-}(i,j) \cdot \Delta_x^-(i,j) \\ F_{y^+}(i,j) &= k_{y^+}(i,j) \cdot \Delta_y^+(i,j) \\ F_{y^-}(i,j) &= k_{y^-}(i,j) \cdot \Delta_y^-(i,j) \end{aligned} \tag{48}$$

The net force in the x-direction and y-direction respectively at the node (i, j) become then:

$$\begin{aligned} F_x(i,j) &= F_{x^+}(i,j) - F_{x^-}(i,j) \\ F_y(i,j) &= F_{y^+}(i,j) - F_{y^-}(i,j) \end{aligned} \tag{49}$$

After the calculation of the forces in each direction, the total spring force is then given by the sum of the net forces from Equation 49.

3.2.5 Excitation

The excitation is designed to be an external force distribution applied to the membrane surface. This force causes localized deformation that starts the wave propagation and vibrational response of the system. In the model, the excitation is implemented as a spatially localized and temporally shaped force that is applied to the mass-spring grid.

First, a strike position is defined within the boundary of the circular membrane, which is applied at a specific radial distance from the center of the grid. If N represents the number of the points in the grid, the strike position $(x_s,\, y_s)$ is computed as the following:

$$x_s = \left\lfloor \frac{N}{2} + r_{\text{strike}} \right\rfloor, \quad y_s = \left\lfloor \frac{N}{2} \right\rfloor \tag{50}$$

where r_{strike} is the radial offset from the center [13]. Since the modes in the membrane depend on the radial position of the excitation, it is useful to have control over the impact location. In addition, assigning different strike positions would affect the sound character of the generated sound.

After the strike position is defined, the spatial shape of excitation is modeled to determine which membrane modes are excited and how strong they are. It also represent how the strike force is distributed across the surface of the grid at the moment of impact. Different types of spatial excitations can be considered, and the impulse excitation is one of them. This type of excitation applies the force only at a single point in the grid corresponding to the strike location, with zero contribution on the other points.

$$f_{\text{excite}}(x,y) = \begin{cases} A, & \text{if } (x,y) = (x_s, y_s) \\ 0, & \text{otherwise} \end{cases} \tag{51}$$

Nevertheless, applying only a spatial excitation such as the impulse, which is over a single time step, would not be completely sufficient. In a practical situation, the contact between the drumstick and the membrane lasts several milliseconds. This contact time shapes the energy transfer in addition to affecting the transient characteristics. With respect to the model in this study, a temporal excitation envelope has been implemented to determine how the input force is applied over time. In signal processing, this is done by shaping the signal over several samples rather than only the first time step [14].

The excitation envelope for the model is implemented using a discrete-time Gaussian expression:

$$e(t) = \exp\left(-\frac{(t-\mu)^2}{2\sigma^2}\right) - \frac{1}{T}\sum_{t=0}^{T}\exp\left(-\frac{(t-\mu)^2}{2\sigma^2}\right) \tag{52}$$

where:

- $e(t)$ is the envelope at discrete time index t,
- μ is the center of the pulse,
- σ controls the duration of the envelope,
- T is the total number of excitation samples.

The spatial excitation is then scaled by the temporal envelope to define the excitation force in both space and time, so the input force has a shape and duration. Figure 6 below shows how the excitation envelope is applied.

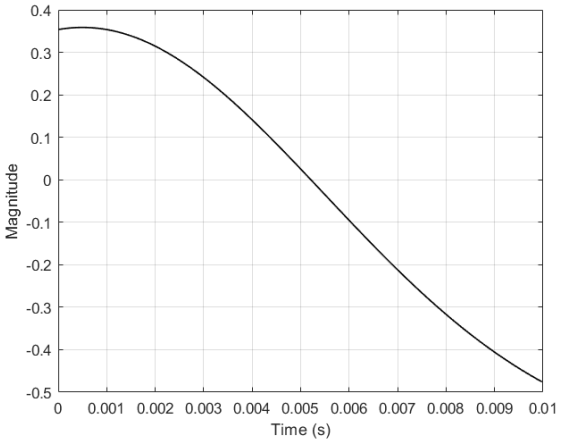


Figure 6: Excitation envelope used to modulate the spatial strike force in time, applied to the drumhead model to shape the temporal profile of the input.

Another technique is added to simulate the bounce of the strike for richer articulation. This is done by scaling the excitation force with a gain coefficient, then adding it to the displacement (u) as follows:

$$u(x, y, t) \leftarrow u(x, y, t) + \gamma \cdot f_{excite}(x, y) \tag{53}$$

3.2.6 Computation

At this stage, the physical state of the mass-spring drum model will be updated in a simulation loop over discrete time intervals. This process will compute the local acceleration of each mass which is represented as a point on the circular grid created. The calculation of acceleration is followed by updating the velocity and displacement respectively. The distributed masses interacts with the developed stiffness, damping, and external excitation models, contributing in a dynamic evolution of sound wave patterns that simulate the sound of percussion. The mechanism of the simulation will be explained in detail.

As mentioned in Section 3.2.5, an external force is applied over a determined area on the membrane grid which is modulated by the excitation envelope. This excitation is active for certain duration. This external excitation force is computed as follows:

$$f_{ij}(t) = \Phi_(i.j) \cdot e(t) \tag{54}$$

where $\Phi_{i,j}$ is the spacial excitation, and $e(t)$ is the envelope calculated from 52.

After the excitation force, the velocity Laplacian is computed to represent the local curvature in the membrane's motion. This is done by using a 2D discrete convolution (see equation 44)with a finite-difference Laplacian kernel to estimate how sharply the velocity field changes around each grid point by comparing its value to the nearest neighbors.

Following this, the damping force acting on each point is computed, and it is frequency and amplitude dependent. After separating the velocity field into low- and high-frequency components, different scaling factors are used to damp these components separately. Including the additional damping terms proportional to displacement magnitude and velocity curvature, the full damping process at each node is given by

$$d_{i,j}(t) = \gamma_{\text{low}} \mid v_{\text{low},ij} \mid + \gamma_{\text{high}} \mid v_{\text{high}}, ij \mid + \gamma_u \mid u_i j \mid + \gamma_{\nabla^2} \mid \nabla^2 v_{ij} \tag{55}$$

Next, each mass interacts with its four immediate neighbors through nonlinear springs. The displacement differences mentioned in equation 46 are used to compute the dynamic stiffness of each spring based on the local extension or compression using equation 47. Eventually, the stiffness forces are used to determine the net force acting on the mass.

With the damping and stiffness forces are calculated, the acceleration is computed using Newton's second law as follows:

$$a_{ij} = \frac{f_{\text{ext},ij} + F_{\text{spring},ij} - d_{i,j} \cdot v_{ij}}{m_{ij}} \tag{56}$$

The resulting acceleration is integrated over time using Euler method to update the velocity and displacement of each mass node:

$$\begin{aligned} v_{ij}(t+1) &= v_{ij}(t) + \Delta t \cdot a_{ij}(t) \\ u_{ij}(t+1) &= u_{ij}(t) + \Delta t \cdot v_{ij}(t+1) \end{aligned} \tag{57}$$

After the update, special boundary conditions are applied, where the rim of the drumhead is determined by a binary mask and reflects energy using a reflection coefficient R. This process modifies the rim values to represent the sound reflection at the edge of the membrane:

$$\begin{aligned} v_{\text{rim}}(t+1) &= R \cdot v_{\text{rim}}(t+1), \\ u_{\text{rim}}(t+1) &= R \cdot u_{\text{rim}}(t+1) \end{aligned} \tag{58}$$

This mechanism simulates the effect of the drumhead mounted on a shell to achieve energy reflection and decay near the boundary. Additionally, to simulate the acoustic coupling with the air inside the drum cavity, a simple resonator model is implemented using a decaying sinusoidal term added to the primary displacement. this coupling is inspired by Helmholtz resonator behavior and adds to the low-frequency response of the system:

$$u_{\text{total}}(t) = u(t) + A_0 e^{-\lambda t} \cdot sin(2\pi f_{res} t) \tag{59}$$

Figure 7 below illustrate the procedure implemented in the simulation loop. It gives a clear flow of the operations within the loop to compute the updates displacement of the drumhead.

By running the simulation loop with a specified duration, the displacement at each time step is restored to generate the sound output. The displacement values are also mapped and saved to be plotted in the results as waveform over time as well as frequency response plots. For the results, different sounds with their plots over time and frequency are generated in terms of four different parameters at three different values each. These parameters are the size, tension, damping, and membrane thickness.

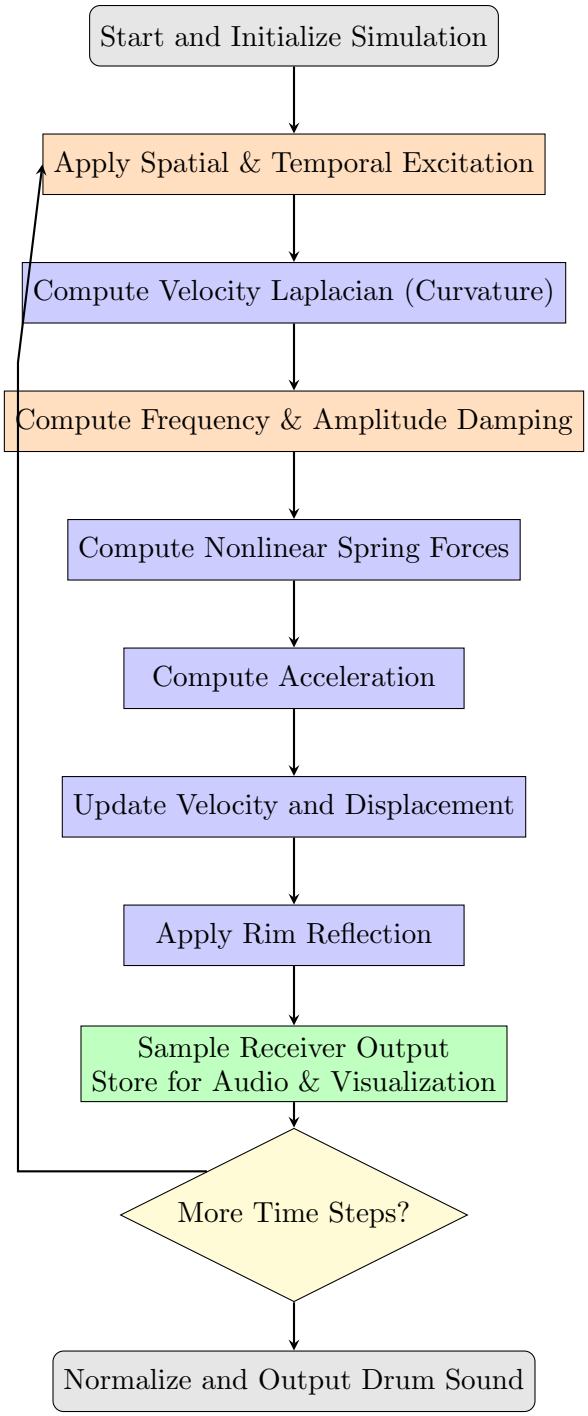


Figure 7: Flowchart of the simulation loop showing the main computation phases for membrane dynamics, damping, excitation, rim reflection, and sound generation. Color-coded nodes distinguish physical modeling (blue), signal processing (orange), and output operations (green).

3.3 Real Time Implementation

Transitioning from the detailed physical model described in the previous sections to a responsive, real-time instrument required a fundamental reconsideration of computational priorities and structural design. This section details the process of adapting the offline model into a performable digital instrument that maintains the essential acoustic characteristics of a drum while operating within the strict computational constraints of real time audio processing.

Real-time audio synthesis presents unique challenges that differ significantly from offline sound generation. Processing must complete within strict latency windows, typically under 10 milliseconds to maintain seamless audio output and responsive performance. Parameters must update smoothly without introducing artifacts, and the system must remain stable across the entire parameter space accessible to users.

The implementation described follows an approach that prioritizes both computational efficiency and musical expressivity. Beginning by examining the available platforms for real-time synthesis and their advantages. Then detail the specific adaptations made to the physical model, including the transition to a polar coordinate system, optimized integration methods, and parameter mapping strategies designed for intuitive musical control. Additional features developed specifically for the real-time implementation, such as a physically-informed snare wire simulation and pitch bend effects, are then presented. Finally, the validation methodology used to ensure that despite necessary simplifications, the real-time model maintains the essential acoustic characteristics identified in the Offline model.

3.3.1 Platforms for Real-Time Synthesis

To implement the physical drum model in real time, the choice was made to use two complementary approaches: MATLAB's Audio Toolbox with plugin generation capabilities and the open-source VCV Rack platform. VCV Rack proved particularly valuable as a development environment due to its modular synthesis architecture that closely resembles hardware Eurorack systems.

VCV Rack offers several advantages for the implementation:

- **Accessibility**: Free core platform
- **Expandable Ecosystem**: Support for third-party developers
- **Development Framework**: Provides a tutorial for creating and distributing custom modules
- **Cross-Platform Compatibility**: Available across Windows, macOS, and Linux
- **Hardware Integration**: Capable of integrating with physical equipment via audio/MIDI interfaces

The platform's real-time interaction capabilities and standardized parameter interface make it ideal for translating the research into practical instruments that musicians can incorporate into their workflows. Additionally, VCV Rack provides comprehensive setup documentation

without requiring external frameworks, simplifying the development process compared to other audio plugin frameworks.

3.3.2 Implementation Approach

The implementation mainly followed two stages:

1. Converting the developed offline model scripts into a MATLAB class structure and using the AudioPlugin toolbox to make a VST Plugin
2. Translating the MATLAB class to C++ for integration with VCV Rack

This dual approach allows to refine the model while optimizing for performance. A critical consideration for real-time audio processing.

The initial development in MATLAB will provide a structured environment for real-time audio processing and parameter management. The subsequent C++ translation for VCV Rack will maintain the same physical modeling approach while optimizing for performance. The C++ implementation also enables a more sophisticated user interface and MIDI capabilities, resulting in a more versatile final product.

3.3.3 Physical Model

While developing the initial MATLAB class script some alterations have been made to the offline model. By opting for a script that still represents the offline model but in a real time situation. Some features needed to be reduced to ensure stability during real time processing and fast calculations while still keep the integrity of the sound.

The first optimization stage was identify the features that made to little effect on the sound and scale the model down to a point where it is fast enough to process the calculations in real time. By doing so the features such as air and shell coupling, shell resonances and adjustable membrane thickness was removed.

Secondly the real-time model represents a drum membrane discretized into a polar coordination system. The choice was made to integrate this setup since it was capable to reproduce drum sounds with fewer grid points by still maintaining key drum behavior such as:

- Dispersive wave propagation across the membrane
- Modal coupling between vibration patterns
- Timbre variations based on strike position
- Realistic evolution of overtones with different decay rates

Lastly the real time model has a changed strike algorithm that is reacting to the MIDI input and maps its velocity. An input threshold and a debounce algorithm set to 20 ms to avoid unwanted re-triggering was also used to prevent unwanted secondary strikes while playing on a MIDI keyboard.

3.3.4 Added Features

The first new feuture introduced in the real time model is the implementation of snare wire response which utilizes a mass-spring approach to recreate the snare wire response. Drawing on principles established by Bilbao [15], the model positions multiple virtual wires across the membrane of the drum, with positions that is normalized to the radius. Each wire is modeled as a point mass with associated displacement and velocity states that evolve over time. When the main drum membrane vibrates, these vibrations are transmitted through a coupling factor. The membranes movement is then sampled at each wire's position, and the velocity of this movement generates forces that act upon the wires. The implementation uses the absolute value of the head velocity to ensure bidirectional response. Wire-head collisions are handled with a simple bouncing mechanism. When a wire's displacement exceeds a threshold value, the system registers a collision, contributing to the overall wire response signal proportional to the displacement amplitude, and then simulates bouncing by reversing the wire's velocity with a 15 % energy loss factor. The noise component stems from a random noise generation using chaotic logistic mapping, generating white noise. The physical response of the wires which acts as an amplitude modulation for the noise component and the collision acts as a dynamic gate, giving a snare like sound.

The second new feature is the pitch bend effect. The pitch bend effect in the real time model mimics the frequency shift that occurs in real drums immediately after being struck with a greater force. The pitch bend effect increases the tension during maximum displacement temporarily raises the pitch before returning to normal. The code implements this by modulating the membrane tension based on parametric adjustment. Upon impact, a bend factor temporarily increases the effective tension, which gradually decays over approximately 100 ms following an exponential curve. This creates the characteristic downward pitch glide that contributes significantly to the perceived realism of percussion sounds.

3.3.5 Time Evolution and Integration

For time evolution, the choice was made to implemented the Verlet integration scheme instead of the Euler method found in 56 as described by Jakobsen (2001) [16]. This second-order numerical method can be expressed as:

$$x(t + \Delta t) = 2x(t) - x(t - \Delta t) + a(t)\Delta t^2 \tag{60}$$

Where $x(t)$ represents displacement at time t, and $a(t)$ is the acceleration derived from forces acting on each grid point. The Verlet integration which is commonly used in game development offered some advantages for the membrane simulation:

- Better energy preservation compared to first-order methods, crucial for accurately modeling oscillatory systems
- Implicit velocity handling, reducing memory requirements
- Numerical stability with fixed time steps
- Minimal numerical drift over extended simulation periods

The acceleration term is calculated by summing all forces (spring and damping) acting on each grid point and dividing by the point's effective mass, with special handling for points near the center to account for the polar coordinate system's singularity.

3.3.6 Real-Time Processing

To ensure responsive low-latency performance, the model implements a block-based processing. Audio is processed in small blocks, with parameter changes managed between blocks to prevent artifacts while allowing continuous adjustment of the drum's properties.

The memory management follows real-time best practices:

- Pre-allocated data structures for the grid state
- Avoidance of dynamic memory operations during audio processing
- Careful buffer management to prevent audio dropouts

Strike excitation is modeled as a Gaussian-shaped force profile applied at the specified strike position by the MIDI signal, creating a localized initial displacement that propagates through the membrane according to physical principles. The time step is fixed according to the sample rate, with model parameters scaled to ensure stability under the Courant-Friedrichs-Lewy (CFL) condition.

3.3.7 Validation Methodology

To validate the implementation, a comparison of the modal frequencies emerging from the physical model with theoretical frequencies predicted by Bessel function analysis was performed. This validation process includes:

1. Analyzing the spectral content of synthesized drum sounds
2. Identifying prominent resonant peaks
3. Comparing these frequencies with theoretical values derived from Bessel function roots

The primary objective of this validation is to assess the model's ability to reproduce the fundamental modal characteristics of circular membranes, rather than to replicate specific drumming styles or tuning preferences. While drum tuning varies significantly across genres—from high-tuned jazz configurations to low-tuned modern gospel setups—this study focuses on validating the underlying physics of membrane vibration. The adjustable parameters in the real-time implementation facilitate comparison with actual drum recordings from the following set of drums:

- 13" tom (with and without resonance head)
- 16" floor tom (with resonance head)
- 14" snare (with engaged wires)
- 22" kick drum

These drums were recorded with standard medium-tension tuning to capture a middle ground, using double-ply Remo Pinstripe drumheads. This selection provides a range of membrane sizes and configurations to test the model's adaptability across different drum dimensions, while the consistent tuning approach ensures that validation focuses on the model's capacity to capture modal behavior rather than genre-specific tonal characteristics. The goal is to demonstrate that the physical model correctly reproduces the theoretical modal structure inherent to circular membranes, which forms the foundation for accurate drum synthesis regardless of specific tuning preferences.

Accelerometer measurements were not used in this study, as the acoustic response captured via microphone provided sufficient spectral information for the modal frequency validation objectives and does not cause any disturbances in the modal behavior pf the membrane.

The recordings of these drums were conducted by placing a condenser microphone 50 cm from the drumhead, pointing at the middle of the membrane and recording in a DAW as seen in Figure 8.

Figure 8: Setup for recording drums.

Validating the physical model's accuracy provides further insight into the characteristics of the simulated drum. A modal analysis comparison is made between the simulated drum and theoretical predictions based on the mathematics of circular membranes. For an ideal circular membrane with fixed boundary conditions, the theoretical modal frequencies follow a pattern described by the Bessel functions—mathematical expressions that naturally arise when solving the wave equation in cylindrical coordinates. These theoretical frequencies are given by:

$$f_{mn} = \frac{c}{2\pi R} \cdot \lambda_{mn}$$

Where:

- f_{mn} is the frequency of the mode with m nodal diameters and n nodal circles,
- c is the wave propagation speed in the membrane,
- R is the membrane radius,
- λ_{mn} is the n-th root of the m-th order Bessel function of the first kind.

This setup is designed to capture relatively natural sound, avoiding near field effects and limiting the presence of room reflections compared to the direct sound.

4 Results

Since this study consist of two main parts which are the generation of the drum sound from the Offline model as well as the real-time implementation, the results are presented to show the aspects and characteristics of the sound created from both models.

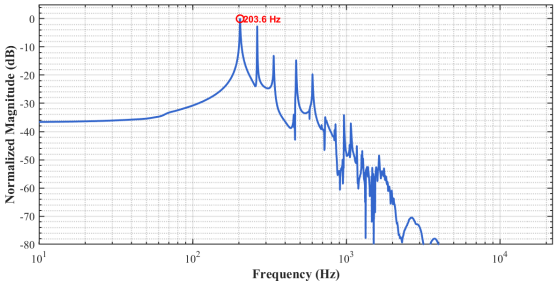
4.1 Offline Sound Generation

To show the results of the Offline model, the sound is represented by plotting the frequency response and the waveform in the time domain. All frequency response plots in this section are presented between 10 Hz and 20 kHz with a normalized magnitude. Time-domain plots have the amplitude between -1 and 1. Also, the duration for all the plots are two seconds with a 100 millisecond silence in the beginning of the generated sound. This process was chosen to avoid any artifacts when generating the sounds, which can cause a signal clipping.

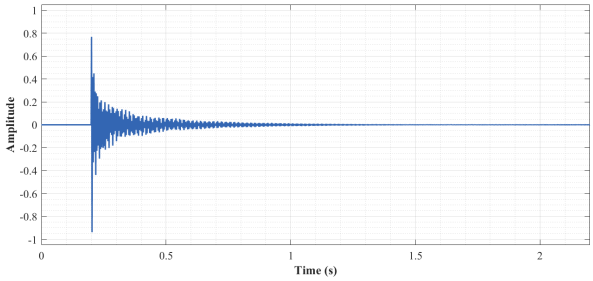
Figure 9 below shows the frequency response and waveform over time of the drum sound generated with variation in the diameter of the membrane. The chosen sizes for the results are 12, 14, and 18 inches. Other parameters are set to be the same for all membrane sizes, where the tension is 500 N/m, the damping parameter is 10^{-4} kg/s, and the thickness is set to be 2×10^{-4} m.

It is noticed that the fundamental frequencies and modal peaks are lower as the diameter of the drumhead gets larger. From 9a 9c 9e, the fundamental frequencies are 203.6, 161.5, and 110.7 Hz for membrane sizes of 12, 14, and 18 inch respectively. Regarding the time-domain plots 9b 9d 9f, it is observed that with larger membranes longer decay is obtained.

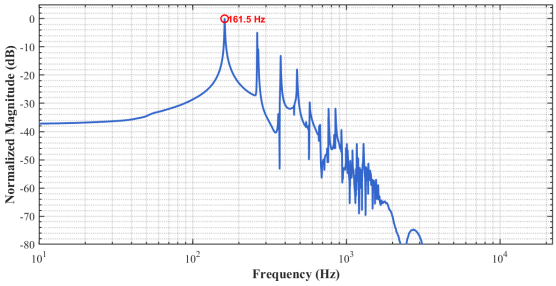
With variation in the size of the membrane, the fundamental frequencies and decays change as expected. Increasing the diameter of the membrane result in increasing the effective wavelength of supported modes. This leads to lower modal frequencies and variations in the modal behavior. In addition, larger vibrating area stores more mechanical energy, which makes the decay time longer. The sound generated also becomes deeper and lower in pitch.



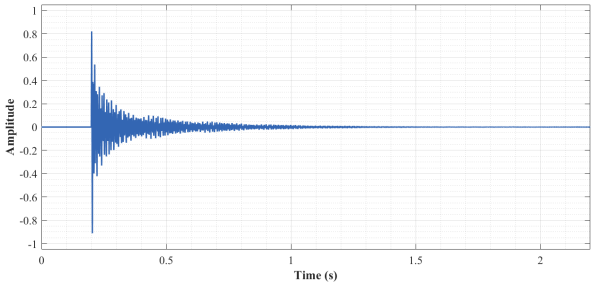
(a) Frequency response for 12-inch drumhead



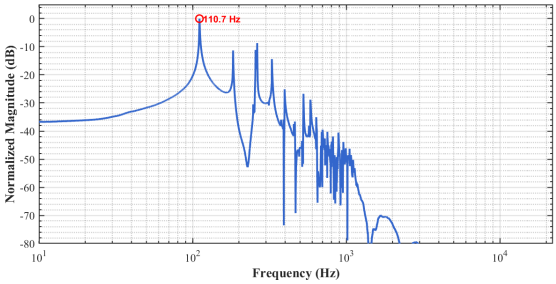
(b) Time-domain waveform for 12-inch drumhead



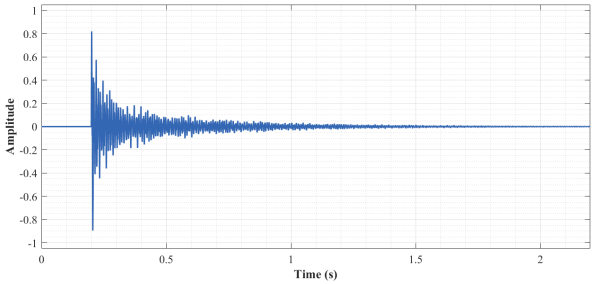
(c) Frequency response for 14-inch drumhead



(d) Time-domain waveform for 14-inch drumhead



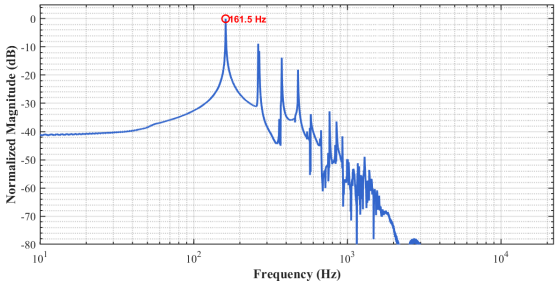
(e) Frequency response for 18-inch drumhead



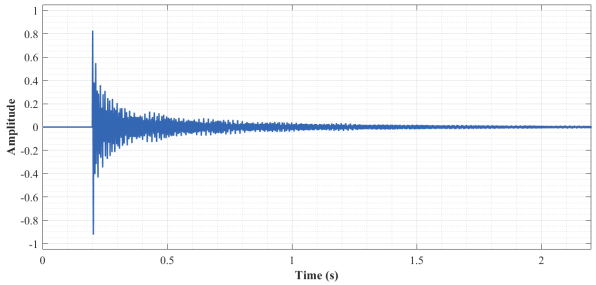
(f) Time-domain waveform for 18-inch drumhead

Figure 9: Frequency response functions (FRFs) and corresponding time-domain signals of the generated drum sounds for three drumhead diameters: 12, 14, and 18 inches. The left column shows the normalized FRFs on a logarithmic frequency axis, highlighting resonant peaks, while the right column displays the decaying waveforms in the time domain. Increasing diameter results in lower fundamental frequencies and longer decay times.

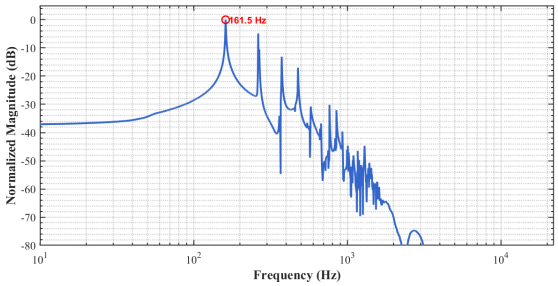
Figure 10 below depicts the frequency response and time domain plots of the generated drum sound that varies with damping values. b is the damping coefficient parameter.



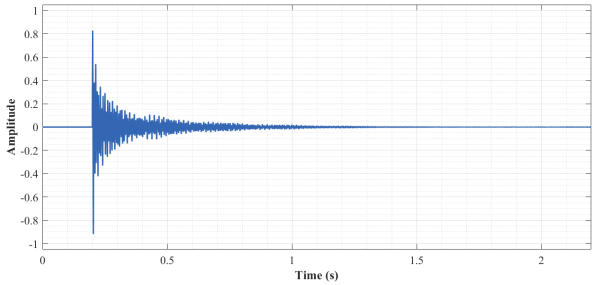
(a) Frequency response for damping coefficient $b = 0.0001$ kg/s



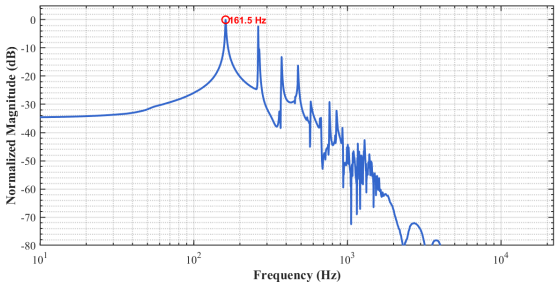
(b) Time-domain waveform for damping $b = 0.0001$ kg/s



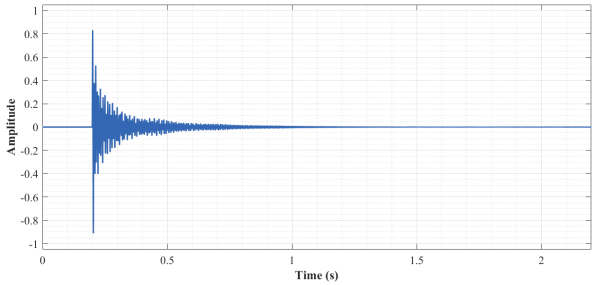
(c) Frequency response for damping coefficient $b = 0.0005$ kg/s



(d) Time-domain waveform for damping $b = 0.0005$ kg/s



(e) Frequency response for damping coefficient $b = 0.001$ kg/s

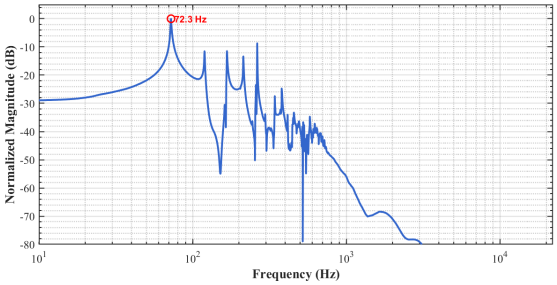


(f) Time-domain waveform for damping $b = 0.001$ kg/s

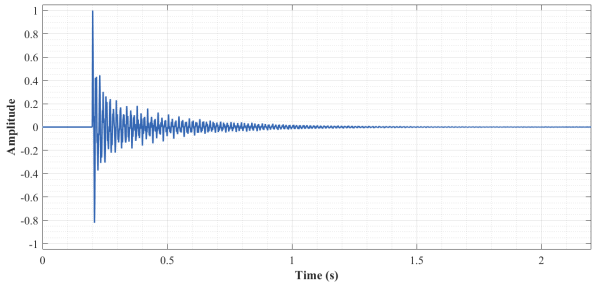
Figure 10: Frequency response functions (FRFs) and corresponding time-domain signals of the simulated drumhead vibrations for three damping values $b = \{0.0001, 0.0005, 0.001\}$ kg/s. The left column presents the normalized frequency spectra on a logarithmic scale, showing the decay of higher modes, while the right column displays the corresponding temporal waveforms. Increasing damping reduces the sustain.

Time-domain plots are shown in 10b, 10d, and 10f. As the damping increases, the sustain of the membrane is generally reduced. In other words, the sound decays faster with increased damping, starting from 10^{-4} kg/s to 10^{-3} kg/s which is expected. On the other hand, it is noticed from the frequency response plots shown in 10a, 10c, and 10e that increased damping do not suppresses the amplitude of all modal peaks. This might have been caused due to the frequency dependent damping having greater effect with less damping.

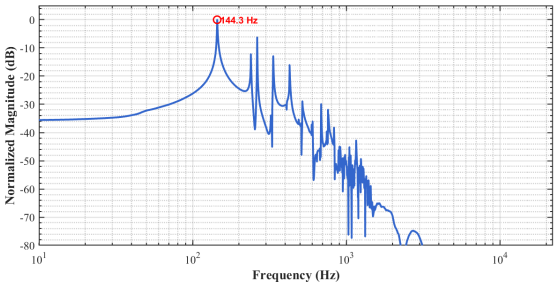
Figure 11 below shows the frequency response and time domain plots of the drum sound generated with tension variations 100, 400, and 700 N/m.



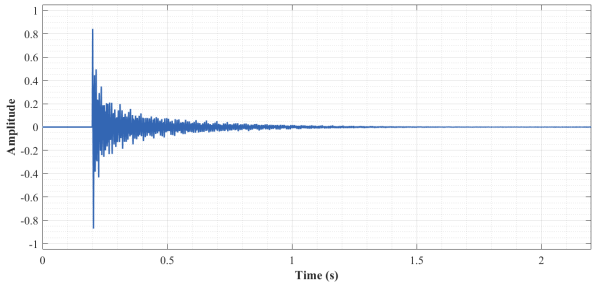
(a) Frequency response for tension $T = 100$ N/m



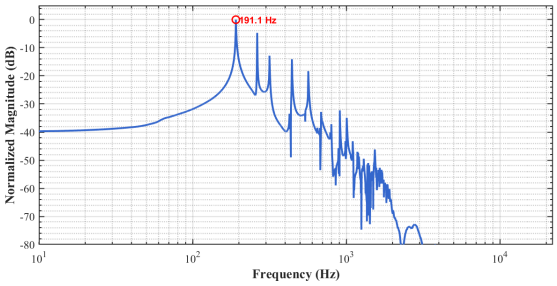
(b) Time-domain waveform for tension $T = 100$ N/m



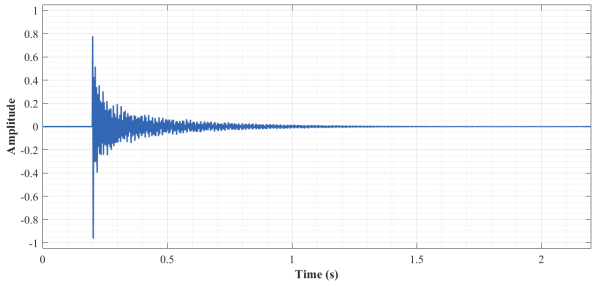
(c) Frequency response for tension $T = 400$ N/m



(d) Time-domain waveform for tension $T = 400$ N/m



(e) Frequency response for tension $T = 700$ N/m

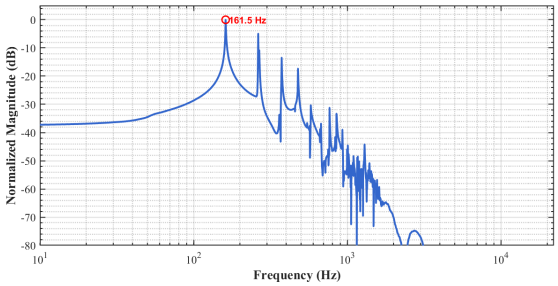


(f) Time-domain waveform for tension $T = 700$ N/m

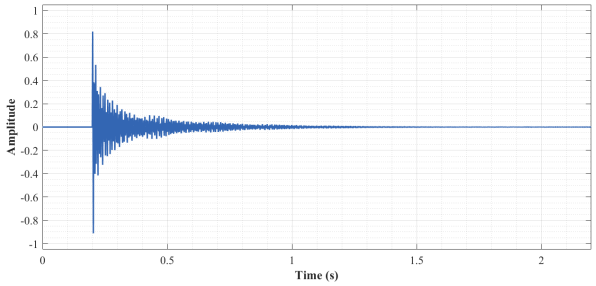
Figure 11: Frequency response functions (FRFs) and corresponding time-domain signals for simulated drumhead vibrations under three membrane tension values: $T = \{100, 400, 700\}$ N/m. The left column shows normalized frequency spectra with resonant peaks shifting higher as tension increases, while the right column displays the corresponding decaying waveforms in the time domain. Increasing tension results in faster wave propagation and slightly faster decay.

Frequency response plots shown in 11a, 11c, and 11e depicts how the fundamental frequency shifts upwards as the tension is set to be higher, which also alters the modal behavior. When comparing the time domain plots for different values of tensions, it is observed how the content of the signal becomes higher in intensity as the tension gets higher, leading to shorter decay as the the membrane becomes tighter. In other words, with increasing the tension of the membrane, the sound becomes brighter the the pitch higher.

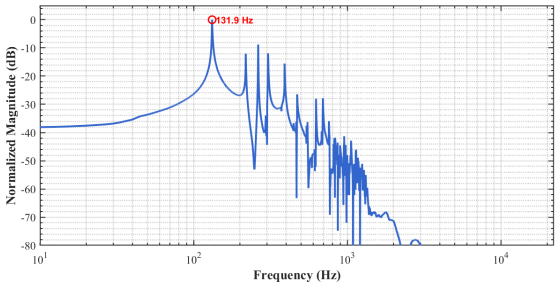
Figure 12 below shows the frequency response and time domain plots of the generated drum sound that varies depending on the membrane thickness.



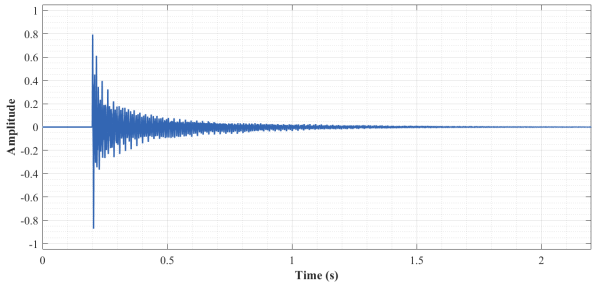
(a) Frequency response for membrane thickness $h = 0.0002$ m



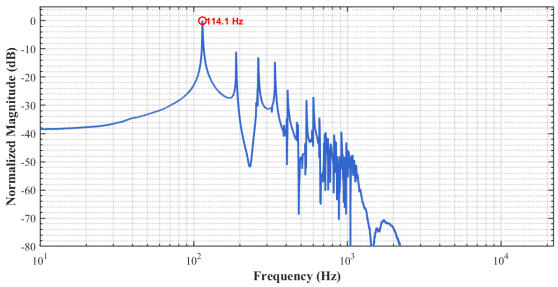
(b) Time-domain waveform for thickness $h = 0.0002$ m



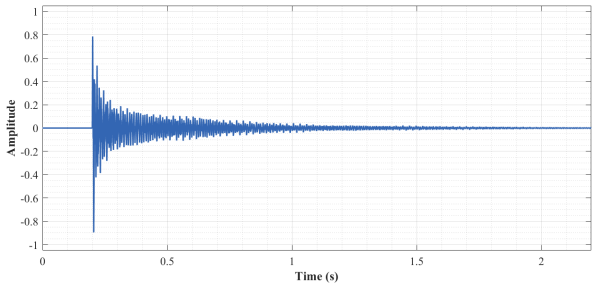
(c) Frequency response for membrane thickness $h = 0.0003$ m



(d) Time-domain waveform for thickness $h = 0.0003$ m



(e) Frequency response for membrane thickness $h = 0.0004$ m



(f) Time-domain waveform for thickness $h = 0.0004$ m

Figure 12: Frequency response functions (FRFs) and time-domain signals of the simulated drumhead vibration under three different membrane thickness values: $h = \{0.0002, 0.0003, 0.0004\}$ m. The left column displays the frequency-domain responses showing resonant mode spacing and attenuation, while the right column shows the corresponding time-domain oscillations. Increasing thickness increases the mass per unit area, which lowers the resonant frequencies and leads to slower transient response and longer sustain.

The frequency response shown in 12a, 12c, and 12e confirms that a thicker membrane leads to increase of the mass. This shifts all modal frequencies slightly down. For the thickness of 2×10^{-4} m, the fundamental frequency is 161.5 Hz, while it is 131.9 Hz for the membrane with a thickness of 3×10^{-4} m and 114.1 Hz for the thickness of 4×10^{-4} m. With a thicker membrane, slower transients and longer sustain are introduced in the signal, indicating sluggish but persistent vibrations as it is shown in 12b, 12d, and 12f. In case of the modal behavior, it is noticed that the change in thickness has a similar effect to tension and size variations but with different timbre.

4.2 Real-Time Results

To validate the real-time physical model, a dual approach is implemented, comparing both acoustic and theoretical properties. This methodology evaluates the model's performance regarding the following perspectives:

1. **Acoustic Validation:** Comparing the frequency characteristics of our model with real drum recordings through frequency response functions, and perceptual assessment.
2. **Theoretical Validation:** Examining the modal frequencies produced by the model against theoretical predictions based on Bessel function roots.

Another aspect of the results is building a real time plugin which is user-friendly. The purpose is to make the plugin as an instrument having elements that is both entertaining and practical in a musical sense.

4.2.1 Model Parameter Mapping and Design

As a result, a final MATLAB plugin is generated and then converted into C++. By using Adobe Illustrator for design, the final module is presented in Figure 13 below.

Figure 13: The Real-Time module in VCV Rack.

The parameters of the module that were chosen to be adjustable are described in Table 3.

Table 3: Mapping between model parameters and their acoustic effects

Model Parameter	**Physical Property**	**Acoustic Effect**
Membrane Tension	Surface tension (N/m)	Primary determinant of fundamental frequency and overall pitch
Membrane Diameter	Physical size (inches)	Sets scaling for placement of the masses and adjusts the grid spacing
Damping Coefficient	Energy dissipation (kg/s)	Controls decay time and sustain characteristics
Strike Position	Normalized radial distance	Influences modal excitation pattern and resulting timbre
Strike Force	Input energy	Determines initial amplitude
Sustain	Expanded control of the decay	Controls decay time and sustain characteristics to extreme values
Snare Blend	Noise generation	Blends in a noise signal that is controlled by the displacement of the membrane and its velocity
Snare Tone	Band-pass filtering	Filtering the noise signal to emphasize certain frequency ranges
Pitch Bend	Tension alteration	Momentary changes the pitch over a short time envelope to mimic hard hits

There are other important parameters like membrane density and membrane thickness described by Fletcher and Rossing [6]. Those could have been selected to be incorporated as adjustable parameters. However, the choice was made to set them as fixed values due the perceptual similarities regarding pitch related parameters like tension and size.

To further expand the user experience and similarities to a real instrument, the module is input-sensitive and do not reset the membrane movement if it gets triggered while still in simulation. This generates randomness in timbre for fast re-triggering. This also interacts with the input velocity of the MIDI instrument which triggers this effect with higher velocity input. Another effect of the velocity from the MIDI signal is output scaling. A lesser velocity will produce less amplitude in the displacement, giving weaker output signal and vice versa.

4.2.2 Spectral Analysis

Comparing the Real-Time model to a real drum, the parameters were adjusted to minimize perceptual differences between the synthesized and recorded sounds. This process resulted in a set of optimized parameters for each drum type, providing insights into how well the model simulates the drum.

By examining the FRF of a recorded drum and a simulation from the drum model in Figure 14 one can notice a spectral difference between the two and a lot of content missing from the simulated drum compared to the real recording. However the perceptual sound is much closer and recognized as a similar drum sound.

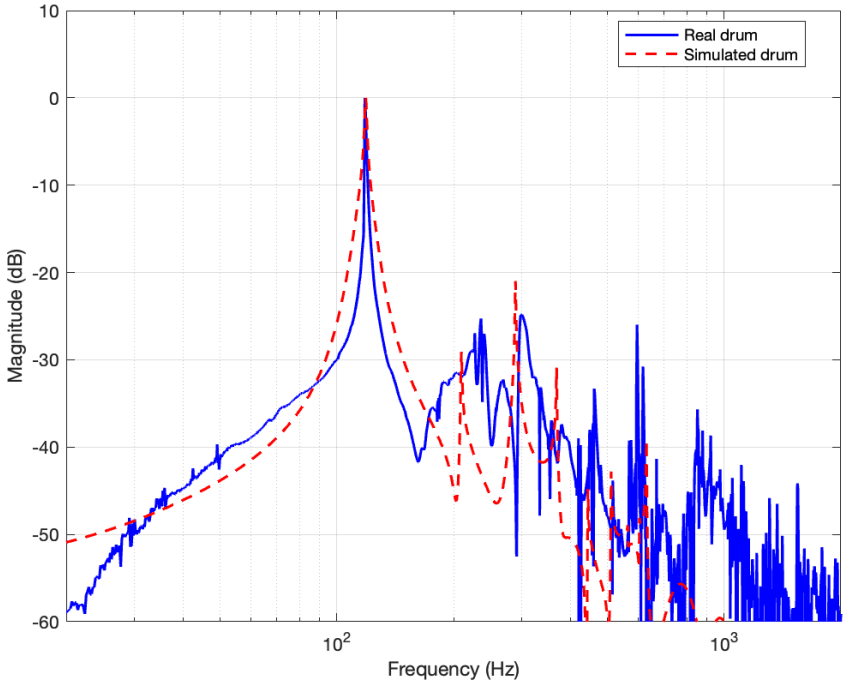


Figure 14: Comparison between real and simulated drum using FRF.

To then validate the FRF with a different approach, a developed MATLAB-based analysis framework was implemented, isolating and comparing the resonant frequencies. The approach begins with identifying the main drum hit in each recording by detecting the maximum amplitude in the signal's energy envelope. From this point, extracting a segment extends approximately 500 ms to capture the resonant behavior where modal frequencies are most apparent. This framework also has an implemented adaptive peak detection algorithm that adjusts its sensitivity parameters based on frequency range. This approach accounts for the natural characteristics of drum sounds, where lower frequency modes typically exhibit broader peaks with greater prominence, while higher frequency components show sharper but less prominent peaks. By dividing the spectrum into multiple bands (20-200 Hz, 200-500 Hz, 500-1000 Hz, and 1000-2000 Hz) and applying detection parameters for each range,

modal frequencies across the spectrum can be identified.

The resulting analysis produced clearer visualizations of frequency spectra with highlighted resonant peaks, allowing direct comparison between real and simulated versions of each drum type.

Figure 15 illustrates the comparison, with both spectra exhibiting similar modal distributions. The fundamental frequency of the real tom appears at 119 Hz, which the model reproduces with minimal deviation. the higher modal frequencies in the 200-500 Hz range show more deviation but align quite well with some of the peaks. It is also important to note that this particular drum had a resonant head which produces its own resonances which is very hard to distinguish from the batter head and its influence on it.

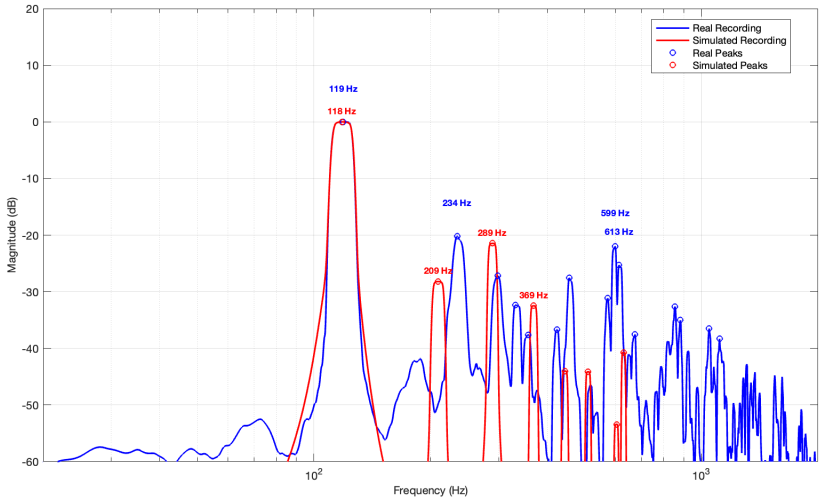


Figure 15: Comparison between real and simulated 13" tom with resonant head.

Using the same drum but without resonant head seen in Figure 16 reveals that even if the real drum is much more related to the single membrane simulation model, the modal behavior is extremely complex of a drum. One can notice that a few more overtones seem to match better and the modal behavior is also clearer in the real drum recording compared to what was seen in Figure 15.

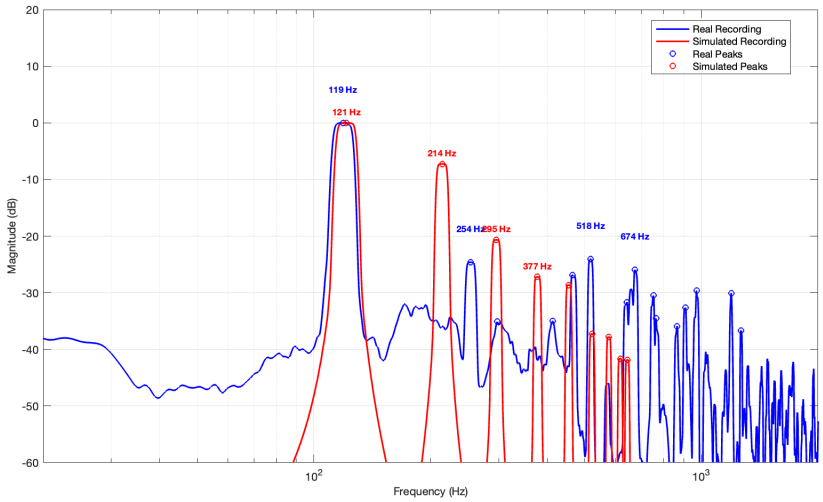


Figure 16: Spectral analysis comparing real and simulated 13" tom without resonance head. The blue solid line represents the real recording spectrum, while the red line shows our simulated version. Key modal frequencies are marked with circular indicators.

For the larger 16" floor tom (Figure 17), The model successfully scaled to reproduce the lower fundamental frequency and altered modal distribution associated with increased membrane size. It is clear from this comparison that the modal distribution and their corresponding frequencies is very close to each other apart from their amplitude after the third partial.

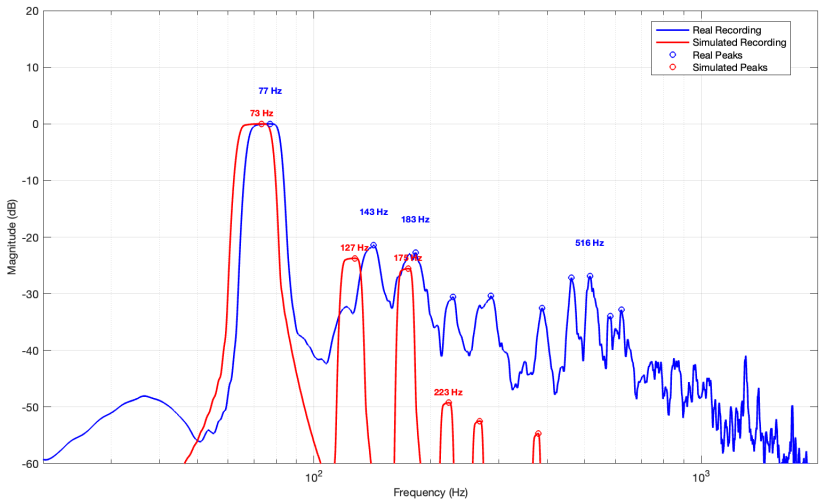


Figure 17: Spectral comparison between real and simulated 16" floor tom. The larger membrane size results in lower fundamental frequency and different modal distribution.

The snare drum in Figure 18 presents unique challenges due to the complex interactions between the membrane and snare wires. The spectral analysis shows that the real snare exhibits a more significant energy in higher frequency ranges (above 1 kHz) compared to other drums. Though not a perfect match, the simulated version captures the essential high-frequency energy distribution that defines the snare sound with modal behavior relatively close to the recording.

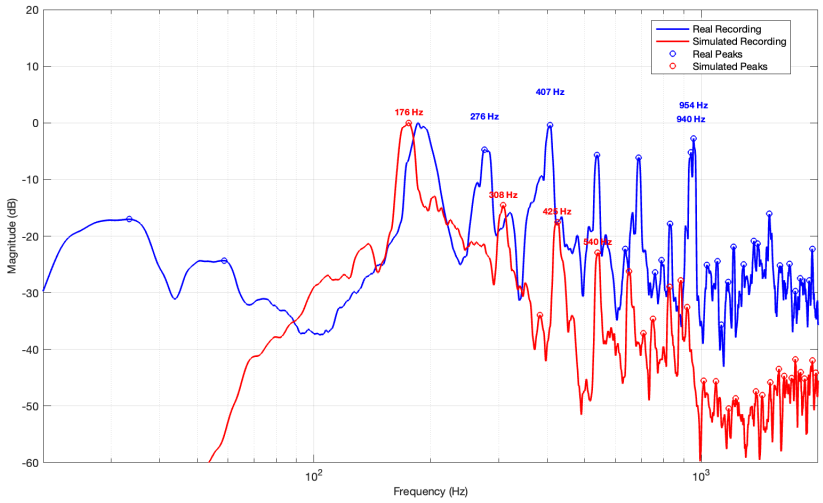


Figure 18: Spectral comparison between real and simulated 14" snare. Note the significant energy in higher frequency ranges characteristic of snare drums.

The kick drum in Figure 19 represents the opposite end of the spectrum with dominant energy below 100 Hz. The model reproduces this low-frequency emphasis, with both real and simulated versions showing similar fundamental modes and energy concentration patterns.

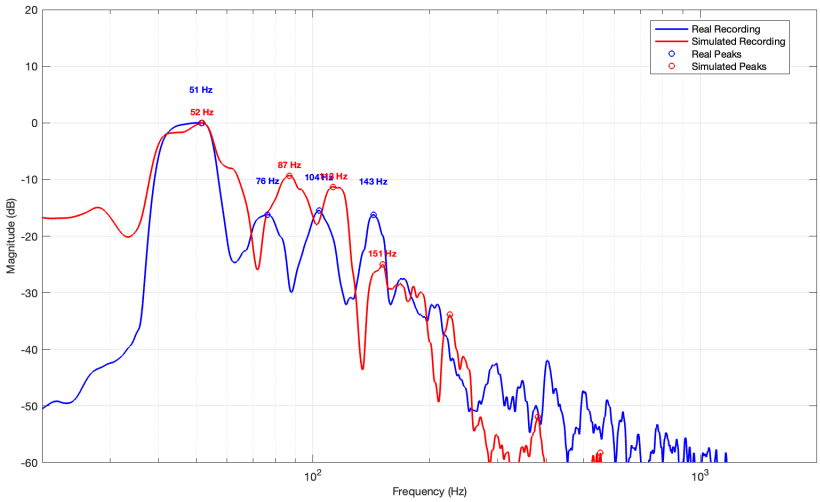


Figure 19: Comparison between real and simulated 22" kick drum. Both spectra show dominant energy concentration in the low-frequency range below 200 Hz.

Across all drum types, the peak detection algorithm identified key resonant frequencies that align with theoretical modal predictions. The close correspondence between these peaks in real and simulated recordings validates the physical model's ability to reproduce drum sounds. Due to the complexity of real drum acoustics, perfect matching of all spectral characteristics remains challenging. However, the model successfully captures the most perceptually significant aspects of each drum type.

4.2.3 Modal Analysis

Figure 20 reveals the harmonic structure of the simulated drum comparing it to the excited modes and their theoretical modal frequencies dependent on the Bessel functions roots.

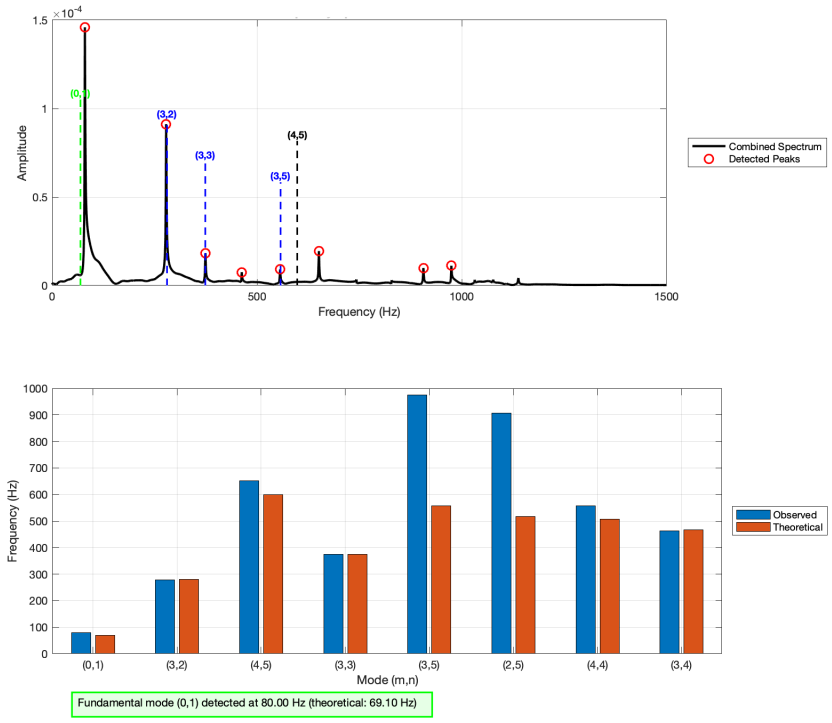


Figure 20: Theoretical difference comparing the simulation of a 14" tom with 270 N/m in tension. The solid black line shows the combined frequency spectrum from multiple sampling points, while red circles indicate detected modal peaks. Dotted vertical lines mark the theoretical frequencies of identified modes with corresponding mode numbers (m,n). Where the green represent the fundamental and the blue the overtones. The second subplot illustrates a bar chart where theoretical and simulated modes are compared side by side.

Notably, the simulation successfully captured the fundamental mode (0,1) at 80 Hz, though with a 15.77 % deviation from its theoretical value of 69.10 Hz, seen figure 21 . This discrepancy is suspected to stem from how the mass-spring system is excited. The higher modes show better agreement, particularly the (3,2) and (3,3) modes with an accuracy of less than 1 % error. Other aspects that might make the simulation deviate is the grid size which acts as resolution. With higher frequencies the modal wavelength gets smaller and can more likely generate a greater deviation from the exact calculated values from the Bessel function much like anti aliasing.

To further visualize and understand how these modes appears physically in the vibrating membrane, Figure 21 illustrates the four most prominent modes in a three-dimensional space.

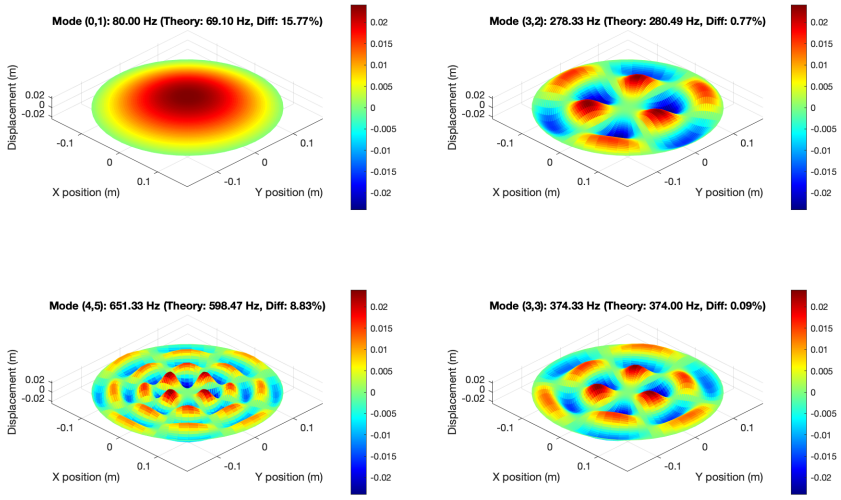


Figure 21: Modal visualization of the drum simulation.

These visualizations reveal the complexity of drum membrane vibration. The mode shapes exhibit symmetries and patterns precisely predicted by the Bessel functions. For instance:

- The (0,1) mode (top-left) shows the fundamental vibration where the entire membrane moves in phase with maximum displacement at the center.
- The (3,2) mode (top-right) displays a pattern with three nodal diameters and two nodal circles, creating six regions that vibrate in alternating phases.
- The (3,3) and (3,5) modes (bottom) demonstrate how higher-order vibrations introduce increasingly complex patterns with more nodal regions.

The modal behavior of the simulated compared to the theoretical values are in detail showed in Table 4

Table 4: Comparison between theoretical and simulated modal frequencies for 14" tom configuration

Mode (m,n)	**Bessel Root**	**Theo. Freq. (Hz)**	**Sim. Freq. (Hz)**	**Error (%)**
(0,1)	2.4048	69.10	80.00	-15.77
(3,2)	9.7610	280.49	278.33	0.77
(3,3)	13.0152	374.00	374.33	-0.09
(3,5)	19.4094	557.74	974.67	-74.75
(4,5)	20.8269	598.47	651.33	-8.83

4.2.4 Informal Perceptual Evaluation [1]

The goal of this thesis was not to create a drum sound indistinguishable from reality, but rather to replicate the fundamental physics of drum behavior. Through physical modeling, the resulting sound occupies a perceptual space between electronic and acoustic instruments, drawing characteristics from both domains. While it may not perfectly mirror a real drum, the mass-spring physical modeling approach enables interactions impossible with other synthesis methods such as sampling, modal synthesis, analog circuits, or delay line models.

The most significant advantage is the model's ability to handle re-triggering during ongoing vibrations, which dynamically alters the sound and creates a much more realistic playing experience. The model also responds authentically to real-time parameter changes—adjusting tension, size, or strike position excites different vibrational modes, and these changes update immediately even while the drum is still ringing. This real-time responsiveness gives the instrument an interactive quality that feels more physical to play.

An unexpected discovery was how the noise generation component, originally designed solely to simulate snare wire behavior, proved to be remarkably versatile. While interacting with membrane displacement for realistic snare effects, this noise system also showed potential for imitating hi-hats, cymbals, hand claps, and other percussive elements. Although these alternative sounds don't achieve the same realism as the primary drum modeling, they demonstrate the broader possibilities inherent in the physical modeling approach.

4.3 Comparison between the offline and real-time models

The following results compares the two drum synthesis implementations. Despite sharing the same physical modeling technique, the main focus of the real-time model is to optimize interactive performance, where the offline model is designed for high-fidelity audio production. The comparison is conducted under two conditions:

- Same Parameters in Figure 22: Both models use identical physical parameters limited to those available in both implementations - drum size, tension, damping, and strike position. However, the models differ in their computational approaches where the real-time model employs a polar coordinate grid system with decreased grid points, while the offline model uses a Cartesian coordinate system with an increased number of grid points.
- Tuned Parameters in Figure 23: The Real-Time model parameters are manually adjusted to match the perceptual characteristics of the Offline model model as closely as possible.

[1] The module can be downloaded for MacOS to VCV-rack with the link in [17].

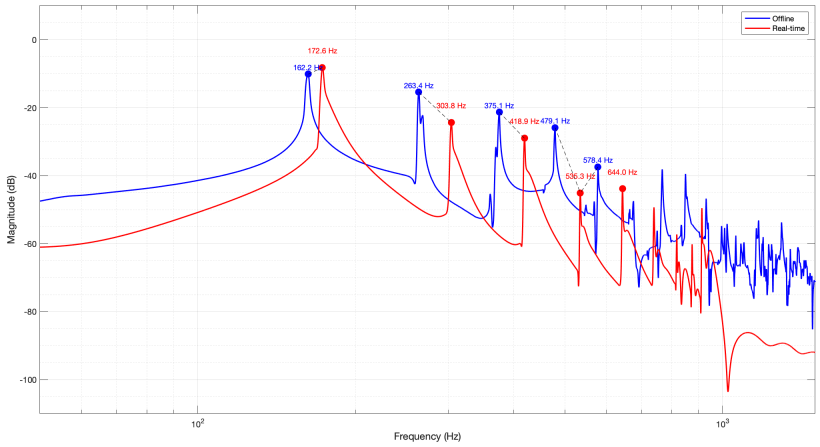


Figure 22: Offline and Real-Time model FRF comparison with same parameters.

The most prominent difference observed in Figure 22 is the shift in the fundamental frequency between the two models. This pitch difference indicates that despite using identical physical parameters, the underlying calculation methods alter the drum's resonant characteristics. Several factors have the possibility to contribute to this difference. The biggest difference between the two is the use of polar vs. Cartesian grid systems. Each system handles boundary conditions and wave propagation differently. The polar system also needs to take account for the non uniform mass distribution and the uneven spring tension between the masses. There is also the possibility that the different time-stepping algorithms and spatial discretization methods can introduce systematic frequency shifts.

While in Figure 23 it is noticeable that both models share the same fundamental mode frequency. Some shifts in the upper partials are present.

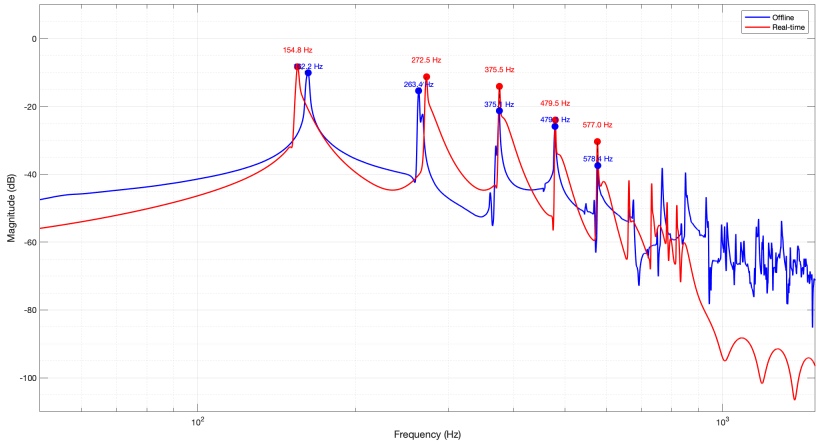


Figure 23: Offline and Real-Time model FRF comparison with tuned parameters.

As seen in Figure 23 both models show similar overall spectral envelopes. However, the real time model exhibits more linear frequency response characteristics. Since the offline model implements both non-linear tension and a more advanced frequency dependent damping technique, it is likely to enhance a more realistic behavior. More closely, it resembles the real drum FRF seen in Section 4.2, having more nuanced overtone structure. Examining the overtone structure further, it is a clearer in the Offline model with the superior grid resolution compared to the real time model. This becomes apparent at 1000 Hz where the Real-Time model is not able to comprehend the modal behavior. Other features like the air coupling and shell resonance of the Offline model is also likely to add realism to the sound.

A consistent observation across both comparisons is that the real-time model produces broader, less sharp spectral peaks compared to the Offline model. This peak broadening phenomenon could arise from multiple interacting factors inherent to real-time processing constraints. The larger time steps required for real-time performance could introduce numerical dispersion that smears energy across adjacent frequency bins, contributing to peak broadening. Another reason could be the frame-based processing inherent to real-time systems tends to average rapid transient behavior of a drum. Despite having a relatively small time steps, the real-time model could result in less defined peak structures compared to the continuous processing capabilities of offline models.

5 Discussion

5.1 Research Questions Alignment

The research questions posed at the beginning of this study can now be systematically addressed, examining how both the theoretical model and real-time implementation contributed to answering each question comprehensively.

The first research question asked: **"How can different physical modeling methods be implemented to model the sound of a percussion instrument, and which one aligns best with expectations?"** Through comparative analysis of FDTD, delay line, modal synthesis, and mass-spring approaches, each method revealed distinct advantages and limitations. The mass-spring model emerged as the most suitable approach due to its optimal balance of physical accuracy and computational efficiency. The spectral analyses in Section 4.1 demonstrate that this approach successfully reproduced the characteristic frequency responses of various drum configurations, establishing clear relationships between parameter adjustments and spectral outcomes. The real-time implementation further validated this choice by showing that even with reduced grid resolution and simplified coupling mechanisms, the mass-spring approach maintained essential modal behavior of drum membranes, as detailed in Section 4.2.

The second research question explored: **"What are the critical parameters that can be used to simulate the modeled sound, and how can they be designed?"** Investigation identified membrane tension, diameter, thickness, and damping as fundamental parameters with predictable effects on frequency response and decay characteristics. The waveform and spectrum analyses in Section 4.1 clearly demonstrate how these parameters systematically affect sound output. The real-time implementation structured the parameters into a musician-centric framework presented in Table 3, where perceptually meaningful controls like "pitch bend" were derived from the physical parameters of the model. The translation from offline to real-time proved essential for creating an instrument that responded naturally in performance contexts.

The third question addressed: **"Is it possible to combine different modeling techniques to achieve better results in simulating the desired sound?"** While the primary approach centered on the mass-spring model, the real-time implementation strategically incorporated concepts from other techniques. Particularly effective was the snare wire modeling, where a hybrid approach combining physical principles with efficient noise generation proved successful. The spectral comparisons in Section 4.2 show that this combination of techniques successfully captured essential timbral characteristics without compromising computational performance.

Finally, the fourth question examined: **"How well does the implemented model perform in a real-time environment and how can it be controlled in an interactive musical setting?"** The transition from the comprehensive theoretical model to the optimized real-time implementation required careful consideration of which physical properties could be simplified without compromising perceptual quality. As detailed in Section 3.3, adopting a polar coordinate system with fewer grid points and the Verlet integration scheme

provided significant efficiency gains. The model's implementation as both a VST plugin and VCV Rack module demonstrated viability in standard music production environments, with the parameter mapping approach described in Section 4.2 providing intuitive control. Perceptual evaluation confirmed that despite necessary optimizations, the real-time model maintained essential characteristics that make it respond like a physical instrument rather than a simple sample playback system.

These findings across both the theoretical model and real-time implementation provide complementary insights into physical modeling of percussion instruments. The theoretical model established the foundation and validated physical principles, while the real-time implementation demonstrated practical application of these principles in performance contexts. Together, they address the research questions from both theoretical and practical perspectives.

5.2 Offline Model

As stated in Section 3.2, this model incorporates the development of the key elements using mass-spring system to capture the complex behavior of the percussion sound, especially a drumhead. The results in Section 4.1 show how the system responds to four physical parameter variations which are the diameter of the drumhead, damping coefficient, tension, and membrane thickness. This response is noted to be consistent with the theoretical expectations and perceptual characteristics of realistic percussion instrument. For example, the variation in the diameter (Fig. 9) show how the fundamental and overtone frequencies shift downward as the diameter increases, which aligns with predictions from Bessel function theory. Also, Larger membranes cause longer decay times as the mass and the storage of energy increases, which is shown i the time-domain plots. This contributes to a deeper and more resonant tone.

The influence of the damping coefficient (Fig. 10) plays a significant role forming the temporal and spectral aspect of the simulated sound. The increase in damping will result in a shorter decay time as well as a darker tone. However, it is observed in the frequency domain the relative prominence of the first overtone while all the modal peaks decay. This can be caused due to the stronger suppression of the fundamental mode. It can be seen as a redistribution of spectral energy that is visible in normalized frequency plots.

The tension parameter discussed in Section 3.2.4 and demonstrated in Figure 11, has a direct and expected behavior where it influences the pitch and the color of the generated sound. Higher tension leads to increase in wave speed, resulting in shifting the modal frequencies up as well as shorter decay times. The nonlinearity introduced in the tension model offers a physical de-tuning feature, making the pitch vary dynamically with displacement amplitude.

Regarding the variation in the membrane thickness, where the frequency response and time-domain plots are shown in Figure 12, a further validation of the model's basis is given. Increasing the thickness of the membrane raises the mass per unit area, reducing modal frequencies and slowing transient response.

The offline model implements a spatial and temporal excitation, described in Section 3.2.5. This excitation model allows to locate the strike at different distances from the center of the drumhead, affecting the modal distribution which leads to variations of the sound character. However, this excitation mechanism remains a simplified approximation since it was hard to model a physical excitation that interacts naturally with the membrane.

In terms of limitations, the model does not incorporate advanced features regarding the shell coupling or air cavity resonance, rather than introducing a simplified calculation of the Helmholtz resonator explained in Section 3.2.6 to be added to the overall sound generated by calculating the displacement values. Otherwise, The whole cavity and the resonating drumhead would be modeled using mass-spring principles, simulating the drum shell to be added to the model. Nonetheless, there was som experimentation with simulating the elements mentioned above, but a negligible effects on the sound was observed.

In general, the offline model provides a physically grounded environment for studying percussion dynamics and behavior, experimenting with different kinds of parameters. The results obtained in Section 4.1 could facilitate the design choices for the real-time implementation discussed in later sections.

5.3 Real Time Model

The spectral analysis comparing real recordings with the real-time model (Figures 15 to 19) revealed further insight into the spectral behavior. While the simulated drums did not perfectly match all spectral components of their acoustic counterparts, they successfully reproduced the most perceptually significant modal frequencies. For instance, the 16" floor tom simulation (Figure 17) captured the fundamental frequency and several prominent overtones with little deviation, despite lacking some complex structures present in the real instrument. This suggests that perfect physical reproduction may be less critical than capturing the characteristic resonant structure that defines particular drum types.

The largest deviation from theoretical values (shown in Table 4) appears at the (3,5) mode, where the simulation differs by 74.75% from theoretical prediction. This substantial discrepancy may result from several factors: modal coupling effects where one mode's excitation influences others, discretization artifacts in the grid-based model, and possible misidentification due to closely spaced theoretical modes in that frequency region. Despite this outlier, the overall agreement between theory and simulation remains encouraging. The close matching of the (3,2) and (3,3) modes (with errors of 0.77% and 0.09% respectively) demonstrates that the model accurately captures membrane vibration physics.

The bandwidth limitations of the real-time model are fundamentally determined by the spatial discretization resolution (number of grid points). Although the model can theoretically represent arbitrarily high frequencies under high tension conditions and small drum size, the practical limitation lies in the finite number of modal overtones that can be resolved above the fundamental frequency. This is also true for the Offline model however, since it utilize a greater grid size this effect is not as prominent.

Modal analysis also reveals important insights about strike position influence on excited

modes. The simulation for the theoretical validation employed a strike position at 20% of the radius from center, which preferentially excites certain modes while suppressing others. This explains why some theoretical modes appear prominently in the spectrum while others remain barely visible. This behavior mirrors real-world playing techniques, where drummers change striking positions to achieve different timbral qualities.

A critical consideration in spectral comparison involves the recording methodology of reference drums. The recordings used microphone placement that captured natural sound, whereas the real-time model generates direct membrane vibration without spatial acoustics. Close-microphone setup might have produced better amplitude matching across modes, particularly in higher frequencies where room effects become more pronounced. However, close-micing introduces complications including near-field effects and potential modal bias depending on microphone position relative to nodal points. Future validation could benefit from multiple microphone perspectives to isolate membrane behavior from acoustic environment effects.

The transition from Cartesian to polar coordinates for the real-time model proved particularly significant. This approach maintained characteristic Bessel-like modal behaviors with substantially fewer grid points—a critical optimization for real-time performance. Modal analysis results confirm that despite this simplification, the model accurately reproduces key resonant frequencies with several modes showing minimal deviation from theoretical values. This validates the approach of strategically reducing computational complexity while preserving physically essential characteristics that define drum sounds. However, theoretical behavior might not represent exact modal frequencies of real drums since calculations don't account for non-uniform tension and mass distribution in the drum head, plus other irregularities that could potentially affect modal behavior and frequencies.

Snare wire simulation implementation presents an instructive case study in hybrid modeling approaches. Rather than attempting to physically model each individual wire which would be computational heavy, the approach of using the membrane's physical state to drive noise generation created a perceptually convincing approximation. This exemplifies how physical processes can inform efficient modeling strategies without requiring exhaustive simulation of every component. The success suggests promising directions for future real-time physical modeling techniques that blend physically accurate components with perceptually-targeted approximations. Similar hybrid approaches could effectively approximate other complex elements like shell resonances and resonant membranes while maintaining computational efficiency.

User interaction with the model revealed another important finding: real-time response to playing dynamics and re-triggering created expressivity levels not typically found in sample-based instruments. The model's continuous response to input velocity and handling of drum hits during ongoing vibration produced variations in timbre and amplitude that contributed significantly to perceived realism. This highlights that in percussion synthesis, dynamic response to performance gestures may be as important as static spectral content.

Successful implementation across both MATLAB-generated VST and VCV Rack platforms demonstrates that physically-modeled percussion can integrate successfully into contempo-

rary music production workflows. The computational optimizations achieved, particularly using Verlet integration and polar coordinate representation, suggest that as processing power increases, more sophisticated physical models will become viable in real-time contexts. This ongoing convergence between offline physical accuracy and real-time performance capabilities presents exciting possibilities for next-generation electronic instruments that combine acoustic percussion's expressive depth with digital synthesis flexibility and control.

5.4 Comparison

Beyond technical spectral analysis, informal subjective listening reveals distinct perceptual characteristics that differentiate the two synthesis approaches. The offline model demonstrates superior attack transient behavior, producing more natural and complex initial strike responses that closely resemble acoustic drum behavior. In contrast, the real-time model exhibits more mallet-like attack characteristics, with somewhat simplified transient responses that, while musically useful, lack the complexity of natural drum strikes. This difference could originates from the excitation radius and time implementation in the algorithm, which remains adjustable but may introduce other implications. The excitation radius for both models was specifically selected to ensure algorithmic consistency and reliable results.

Decay characteristics further distinguish the models, with the offline model producing significantly more natural decay behavior through its frequency-dependent damping implementation. This approach results in a realistic response over time, where different frequency components decay at physically realistic rates. The real-time model, constrained by computational efficiency requirements, employs more uniform damping characteristics that, while stable and predictable, cannot replicate the nuanced decay behavior observed in acoustic drums.

Harmonically, the models present compelling trade-offs. The real-time model generates a somewhat more regular harmonic series, which may prove desirable in musical contexts requiring cleaner, more controlled sounds. However, this regularity sacrifices the natural irregularities and inharmonic content that give acoustic drums their distinctive timbral character. The offline model better preserves these natural harmonic relationships, contributing to its more organic and authentic sound character.

Perhaps most significantly, the offline model possesses a distinctive "airiness" that effectively simulates the acoustic contribution of a drum shell resonating within three-dimensional space. This spatial quality, combined with subtle spectral roughness, creates a convincing illusion of an acoustic instrument existing in real physical space. The real-time model, while lacking this spatial authenticity, compensates with its own character that may prove advantageous in some contexts where hybrid acoustic-electronic aesthetics and performance are desired.

The fundamental distinction between the models lies in their practical application capabilities. Despite certain sonic limitations, the real-time model offers complete interactivity, responding dynamically to input velocity variations and enabling seamless retriggering during performance. It functions as a fully playable instrument capable of expressive real-time musical performance. Where the Offline model, with its computational complexity and high-fidelity processing requirements, cannot currently operate in real-time scenarios. This

exemplifies the classic trade-off between acoustic authenticity and practical musical functionality, where model selection depends on whether the priority lies in achieving maximum sonic realism or enabling interactive performance capability using physical modeling techniques.

6 Conclusion

The purpose of this project was to explore and develop physical models for drum sound synthesis, aiming to achieve acoustic accuracy while maintaining computational efficiency. After evaluating four different physical modeling methods, the mass-spring approach was identified as the most suitable for drum sound modeling. The evaluation considered properties such as realism, parameter control capabilities, and real-time implementation feasibility.

The offline model, implemented in MATLAB, was designed to incorporate advanced physical behaviors including nonlinear tension, frequency- and amplitude-dependent damping, as well as spatial and temporal excitation mechanisms. Results obtained from this model aligned with theoretical predictions and responded appropriately to changes in physical parameters such as membrane size, tension, damping coefficient, and thickness variations.

The real-time version employed a polar coordinate system instead of the Cartesian approach used in the offline model to achieve computational efficiency. This implementation introduced additional features including MIDI-controlled excitation, pitch bend dynamics, and snare wire simulation. The model was successfully deployed both as a VST plugin and as a module within the VCV Rack modular synthesis environment. Spectral and modal analyses validated the model's ability to reproduce the fundamental modal structure of drum sounds, while perceptual evaluations confirmed its expressivity and responsiveness under performance conditions.

Despite these achievements, several limitations were identified, including the simplified excitation mechanism, absence of detailed shell and air cavity modeling, and some deviations from real acoustic recordings in spectral characteristics. These limitations point toward specific areas for future development while demonstrating that physically-based drum synthesis can successfully balance acoustic authenticity with practical computational constraints.

Future research could address these limitations through several promising approaches. The excitation mechanism could be enhanced by incorporating contact time modeling based on strike characteristics, which could provide more realistic attack transients and velocity-dependent behavior. The real-time model's tension implementation currently exhibits overly linear behavior. Developing more sophisticated nonlinear tension modeling would improve the authenticity of pitch variations under different playing dynamics. Additionally, the relationship between size and tension parameters requires refinement to produce more distinctive sonic differences when adjusted.

A particularly promising direction involves extending the real-time model to include secondary membrane simulation for snare drums and similar instruments. This could be achieved through simplified physically-informed filtering approaches rather than full physical modeling, maintaining computational efficiency while adding acoustic complexity. Similar

filtering-based techniques could effectively approximate shell resonance, drum depth effects, and air cavity contributions. Essentially translating the offline model's comprehensive physical behaviors into computationally efficient real-time implementations. This hybrid approach of combining core physical modeling with targeted filtering could significantly enhance realism while preserving the interactive performance capabilities that make the real-time model practical for musical applications.

References

[1] *Eurorack*, https://en.wikipedia.org/wiki/Eurorack, Accessed: 13-May-2025, 2025.

[2] S. Bilbao, *Numerical sound synthesis: Finite difference schemes and simulation in musical acoustics*, https://http://www.wiley.com/, 2009.

[3] D. Morgan, "Mass-spring networks for sound synthesis," M.Sc. Thesis, McMaster University, Hamilton, Ontario, 2008. [Online]. Available: https://macsphere.mcmaster.ca/handle/11375/8700.

[4] Ansys. "What is finite-difference time-domain (fdtd)?" (Jan. 2024), [Online]. Available: https://www.ansys.com/blog/what-is-fdtd (visited on 05/16/2025).

[5] M. Kahrs and K. Brandenburg, *Applications of Digital Signal Processing to Audio and Acoustics*. Boston, MA: Springer, 1998. DOI: 10.1007/978-1-4757-3921-9.

[6] N. H. Fletcher and T. D. Rossing, *The Physics of Musical Instruments*, 2nd ed. New York, NY: Springer, 1998. DOI: 10.1007/978-1-4612-0530-3.

[7] J. O. S. III and S. A. V. Duyne, "Physical modeling with the 2-d digital waveguide mesh," *Proceedings of the International Computer Music Conference (ICMC)*, pp. 268–271, 1995. [Online]. Available: https://ccrma.stanford.edu/~jos/waveguide/Physical_Modeling_2_D_Digital.html.

[8] G. Kelly, *Mechanical Vibrations*, 1st. McGraw-Hill Education, 2005, ISBN: 9780071198929.

[9] J. A. Laird, *The physical modelling of drums using digital waveguides*, Department of Electrical and Electronic Engineering - the University of Bristol, 2001.

[10] S. Willemsen, "Nonlinear Strings Based on Masses and Springs," in *Proceedings of the 26th International Conference on Digital Audio Effects (DAFx)*, Copenhagen, Denmark, Sep. 2023, pp. 1–7.

[11] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Prentice Hall, 1999.

[12] L. Meirovitch, *Fundamentals of Vibrations*. New York: McGraw-Hill Education, 2001.

[13] A. Chaigne and V. Doutaut, "Numerical simulations of xylophones. i. time-domain modeling of the vibrating bars," *The Journal of the Acoustical Society of America*, vol. 101, no. 1, pp. 539–557, 1997.

[14] P. R. Cook, *Real Sound Synthesis for Interactive Applications*. CRC Press, 2002.

[15] S. Bilbao, "Time domain simulation and sound synthesis for the snare drum," *The Journal of the Acoustical Society of America*, vol. 131, no. 1, pp. 914–925, Jan. 2012. DOI: 10.1121/1.3651240.

[16] T. Jakobsen, "Advanced character physics," in *Proceedings of the Game Developers Conference*, CMP Media, 2001, pp. 383–401.

[17] [Online]. Available: https://github.com/albin0e/SpringyDrums/tree/73696606279ba5bed2397aSpringyDrums.