“Singing Glasses” – Digital Adaptation of the Glass Instrument

HAJIM SCHOOL of ENGINEERING & APPLIED SCIENCES, UNIVERSITY of ROCHESTER

Tianwei Jiang, Zilin Zeng, and Michael C. Heilemann

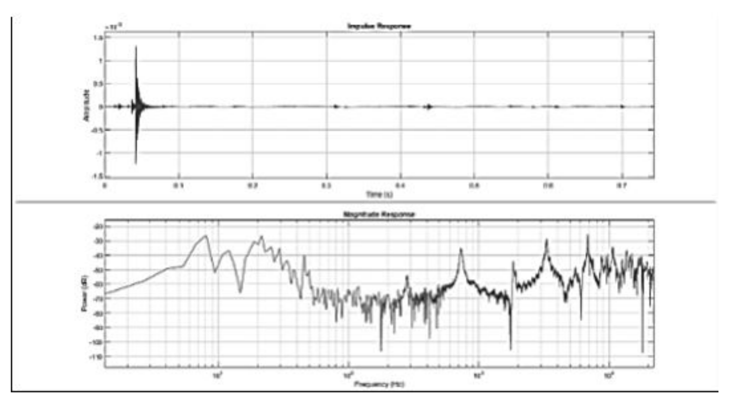
Dept. of Audio and Music Engineering, University of Rochester

Motivation for Research

Rubbing one's finger along the rim of a wine glass produces a unique sound that has been a popular party trick for generations. Though the sound has made its way into popular music recordings, the ergonomics of the instrument make it impractical for widespread use. To make this effect easier to implement, the researchers developed an audio plug-in that simulates the sound of the glass digitally. The researchers have also developed a prototype device that uses a MIDI (musical instrument digital interface) keyboard to spin a particular glass via a DC stepper motor when the appropriate key is pressed. This gives the potential for a single user to generate multiple pitches simultaneously.

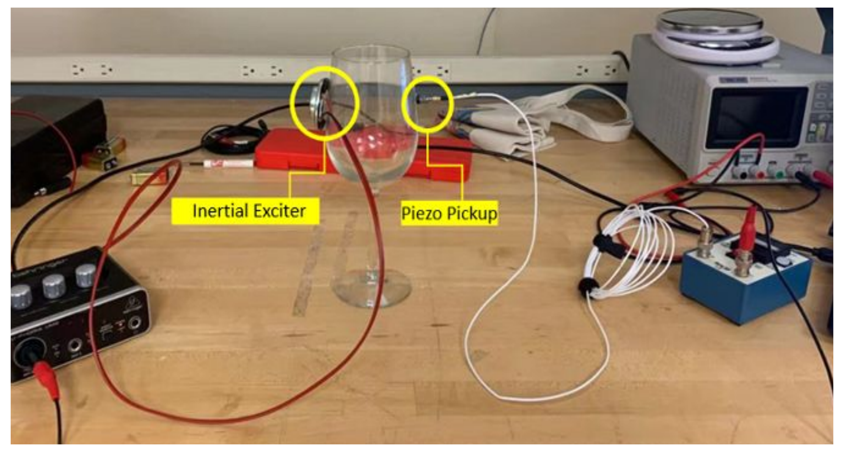
Impulse Response Measurement

Our system strictly followed a Linear Time-Invariant (LTI) system, so the audio effect was achieved by convolving the impulse response of a designed wine glass with the user’s audio input. The default impulse response was measured using the Impulse Response Measurer app from MATLAB, as shown in left figure below.



Default impulse response for an empty universal-shape wine glass

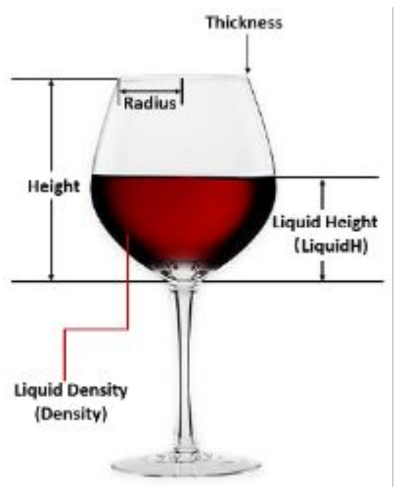
An inertial exciter was attached to the body of the wine glass. We used the Maximum Length Sequence (MLS) method that excited the wine glass with white noises played through the inertial exciter. A piezo pickup was attached to the opposite side of the glass to record the default impulse response from the wine glass due to noise excitation.



Measurement set up for the impulse response of an empty universal-shape wine glass

Equations for Frequency Computation

Users may modify the wine glass impulse response by changing the following parameters: wine glass height, rim thickness, rim radius, liquid height, and liquid density.The default impulse response retrieved was the response for an empty universal-shaped wine glass. The fundamental frequency is given by [1] :



User-controlled plug-in parameters

$$V_o' = \frac{1}{2\pi}\left(\frac{3Y}{5\rho_g}\right)^{\frac{1}{2}}\frac{a}{R^2}$$

where Y is the Young's Modulus for glass, ρ_g is the density of the wine glass, a is the rim thickness of the wine glass, and R is the rim radius for the wine glass.

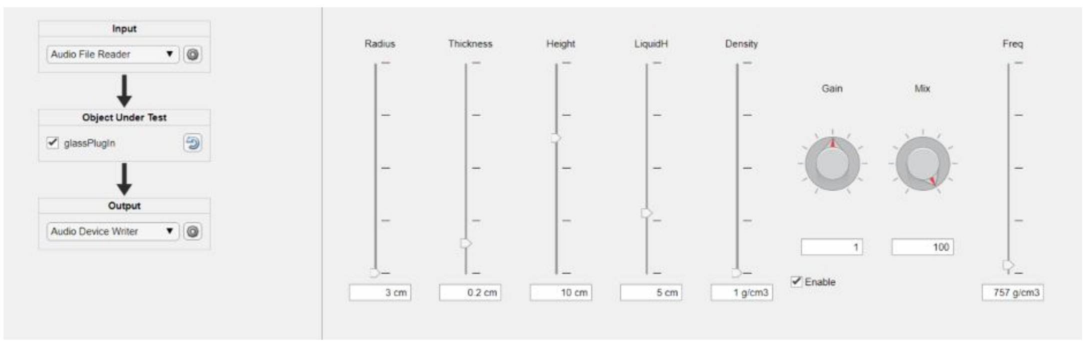
Based on the user's input parameters on the physical properties of a wine glass, we calculated the fundamental frequency v_h for a user-designed wine glass using the following equation [1]:

$$\frac{V_o}{V_h} \approx 1 + \frac{\alpha}{5}\frac{\rho_1 R}{\rho_2 a}\left(\frac{h}{H}\right)^4$$

where α is a value determined by the shape of the wine glass, which typically equals 1.4, ρ_l is the density of the liquid, h is the height of the liquid, and H is the height of the body of the wine glass.

“Glass Reverb” Software Plug-in

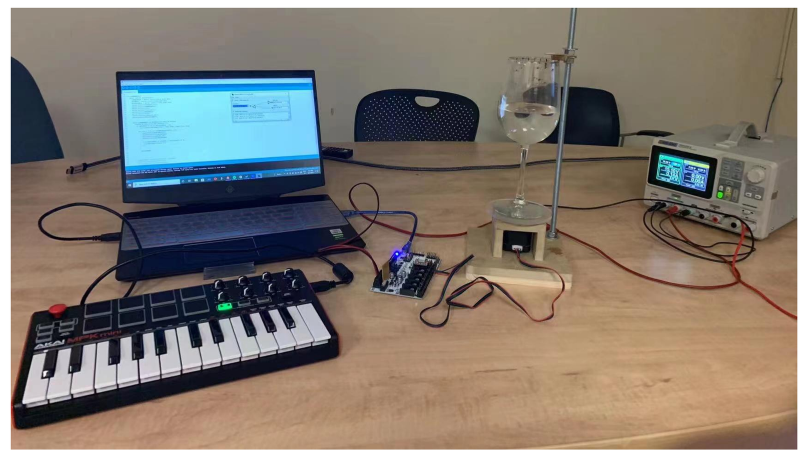
We then modified the default impulse response by changing the parameters for the wine glass. We resampled the default impulse response using that ratio to shift the pitch of the wine glass response based on the user design.



Plug-in GUI configuration

We used MATLAB to design our audio plugin. The plugin was further converted into AU(Audio Units) and VST3 (Virtual Studio Technology 3), which are the standard formats that are compatible with almost all the DAWs(Digital Audio Workstation). The figure on the left shows the Graphical User Interface (GUI) for the “Glass Reverb” plugin.

Glass Harp Instrument



Whole setup for the instrument

A mechanical glass harp instrument is built to enable a single user to play multiple wine glasses simultaneously. A keyboard is used to interface between the user and the set of glasses. A stepper motor rotates each glass when the appropriate key on a MIDI keyboard is pressed. A microcontroller is used to facilitate the communications between the MIDI keyboard and the mechanical motors.

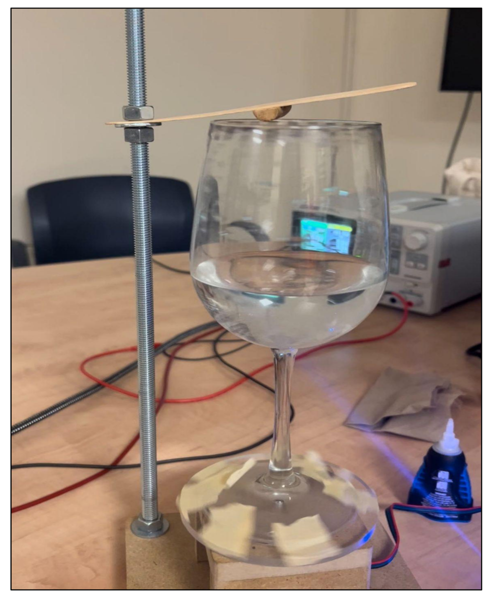
Motor Control

Setup: Midi-controlled motor

The motor control is achieved by using Arduino to send commands to a Mega 2560 rumba board with 6 motor slots, which can be individually programmed to control wine glasses with different water levels. A MIDI keyboard is connected to the user’s PC, which can send commands via a COM (communication port) serial port to control the motors. The MIDI connection is built via a software application called Hairless MIDI. We have embedded our rumba board with a while loop coded with Arduino that constantly checks the MIDI condition from the connecting keyboard.

Once our user presses a key on the keyboard, a note-on MIDI message will be sent to the board for further evaluation. Our Arduino program will figure out what note our user is pressing on and whether that note belongs to one of our six notes. If that evaluation yields a positive result, the rumba board will set the motor to high and thus will turn the corresponding wine glass. After a while, as the user moves the finger off the keyboard, a note-off message will be sent to the board that instantaneously stops the motor.

Mechanical Design



Setup: Motor enclosure and wine glass

An enclosure is designed to hold the motor and an acrylic disk is well cut to fit into the shaft so that the disk can spin with the shaft. The center of the acrylic disk is engraved to have a groove that fits with the bottom of the glass. The enclosure allows the glass cup to spin with the shaft.

Wet cork is used to excite the glass to generate the pitch. We have constant mist design to keep the wood cork in a wet condition. The shelf has a moving chip that can move up and down to provide enough force to the cork such that it can press onto the rim of the glass and excite the glass.

Discussion

This system is currently monophonic, meaning that only one note can be played at a time. Yet, if our users is wants to play chords (a multiple set of notes), a more sophisticated MIDI input system is needed. The next step will be modifying our Arduino code, set up a MIDI buffer, and upgrade our system to polyphonic using a first-in-first-out (FIFO) MIDI parser.

A consistent water level is required to maintain the proper tuning of each note. A more detailed investigation into the lubrication system is required to ensure that the rim maintains the proper level of lubrication to generate a tone without disturbing the water level in the glass itself. The prototypes demonstrated in this work demonstrate a proof-of-concept for a more complex electronic instrument system.

[1] French, A. P. “In Vino Veritas: A Study of Wineglass Acoustics.” American Journal of Physics 51, no. 8 (1983): 688–94. https://doi.org/10.1119/1.13147.