Slapbass synthesis using digital waveguides

A report submitted as part of the course project for MUMT 618
by
Abhiram Bhanuprakash
McGill University
05 December 2018

Abstract— **Slapbass is an electric guitar playing technique which players use to produce a percussive sound. Previous studies have shown that digital waveguide-based synthesis can be used to synthesize this sound. In this course project, an attempt is made to implement two such methods to synthesize slapbass. The "plucked" slapbass sound is successfully implemented. This report documents the observations made and challenges faced during the implementation of these two techniques.**

I. INTRODUCTION

The use of the slapbass technique in electric guitars produces a "brilliant and percussive" sound [1]. Slapbass is a non-linear effect produced by the string hitting the fretboard and bouncing back. The non-linearity arises from the string's displacement being limited by the fretboard.

In the digital waveguide formulation [2], a string is treated as a delay line with the delay being proportional to the length of the string. Since we are sampling the space along the length of the string, if the sampling frequency is f_s $samples/s$ and the speed of sound in the medium of the string's material is c m/s, each sample corresponds to a spatial sampling interval of c/f_s m. Specifically, if $f_s = 44,100$ $samples/s$ and $c = 345$ m/s, this interval corresponds to ≈ 7.8 mm.

If the string is rigidly terminated at both ends, a wave induced at one end of the string will propagate along the length of the string, is reflected at the other end, comes back to the initial end and is reflected back. This repeats at a period of $2L/c$, where L is the length of the string. If there is no loss (ideal case), this periodic motion repeats forever. However, in reality, there are losses in the string due to factors like yielding terminations. In the digital waveguide model, these losses are accounted for using a lumped gain and digital filters (for the frequency-dependent losses).

The wave equation for a lossless vibrating string is given by

$$Ky'' = \epsilon\ddot{y}$$

where K is the string constant, ϵ is the mass density and $y(x,t)$ is the string displacement at a point x on the string at time t. $y'' = \frac{\partial^2 y}{\partial x^2}$ and $\ddot{y} = \frac{\partial^2 y}{\partial t^2}$. The solution of this equation results in two traveling waves, conventionally treated as *right-going* ($+ve$) and *left-going* ($-ve$) waves. At any given time, the sum of the $+ve$ and $-ve$ going waves at a given point results in the string displacement y, i.e., $y_+(t,x) + y_-(t,x) = y(t,x)$.

Coming to the slapbass, there are two kinds of sounds which can be produced by two distinct playing actions: (a) *Plucked* case where the string is pulled away from the fretboard and released to hit the frets, or (b) *Slap* case where the string is slapped by the knuckle of the thumb towards the frets.

We know that the Karplus-Strong algorithm [3] can be used to synthesize a string instrument's timbre. This algorithm and its extensions [4] show that the digital waveguide is a powerful tool to synthesize such instruments. In the following sections, we will see how this algorithm can be improvised to account for non-linear effects like the slapbass.

II. LITERATURE REVIEW

The authors in [1], [5] have successfully used the digital waveguide formulation to implement slapbass synthesis. In both [1] and [5], the authors consider the *displacement* wave, because it is a natural choice in this case as the fretboard is limiting the string displacement. Here, a brief overview of these articles is presented. For details, the reader can refer to the articles.

A. Rank and Kubin method

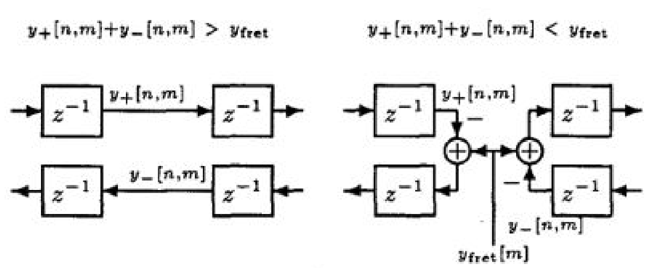

Fig. 1. [1] Left: The normal + and - delay lines in the case when the string displacement does not exceed the fret displacement, Right: The broken waveguide showing the displacement limiting when the string displacement exceeds the fret displacement. In both cases, n and m are the time and waveguide section indices, respectively, and y_{fret} is the fret displacement, $y_{fret} < 0$

Fig. 1 shows the central idea in [1]. When there is no collision with the fretboard, the string is vibrating freely and hence the waveguide updates normally. However, in the case of the displacement exceeding the fret displacement, $y_+[n,m] + y_-[n,m] < y_{fret}$, the waveguide is broken at that point. Treating the collision point as a rigid termination,

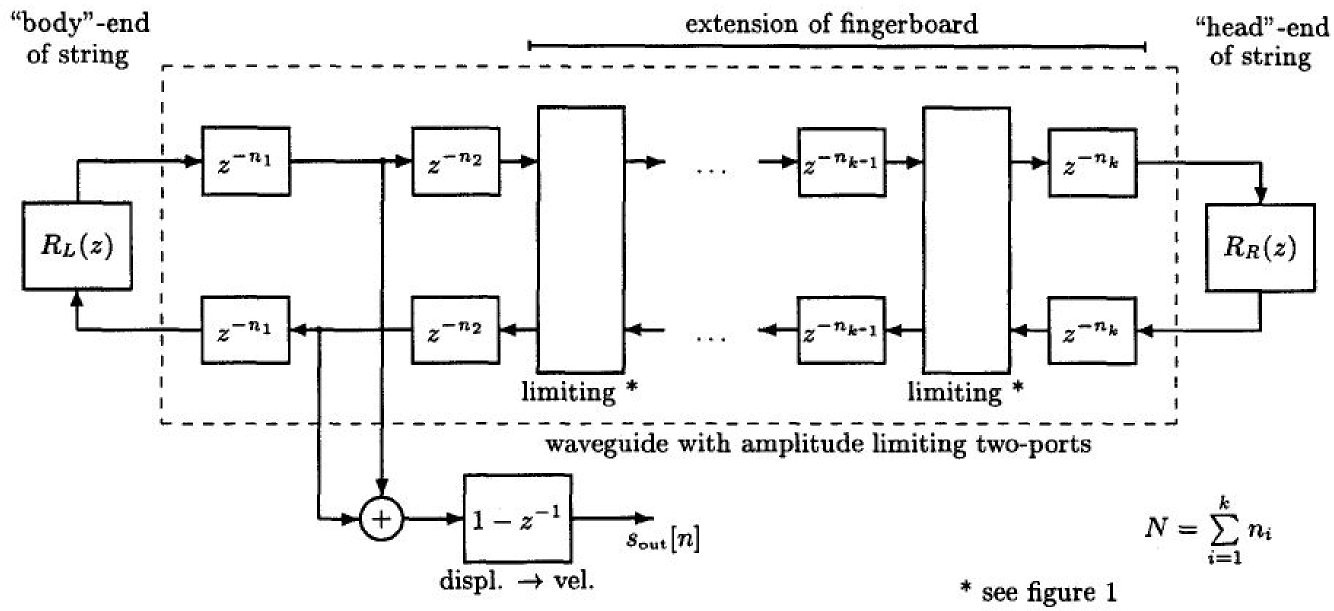

Fig. 2. Waveguide model proposed by Rank and Kubin [1], used in this project. n_is are the individual delay line lengths for the sections between the limiting blocks, for $i = 1, 2, ...k$. n_1 is chosen proportional to the free length of the string.

the two traveling waves are reflected with a sign inversion and the fret displacement y_{fret} is added to each of them to limit the string displacement $y_+[n,m] + y_-[n,m]$ to y_{fret}.

Once the string comes back from the fret, $y_+[n,m] + y_-[n,m] > y_{fret}$, and thus the waveguide is again back to the normal position as shown in the left part of Fig. 1.

B. Krishnaswamy and Smith method

The authors in [5] improvise on the Rank and Kubin method. They recognize that, in [1], during the short collision interval, the waveguide is being split into two sections and the fret distance is added, which makes the +ve and -ve waveguides discontinuous.

Thus, joining the sections back when the string comes back from the fret is somewhat artificial, because we would then be matching two portions of mismatched waveguides together, which is physically unrealistic. They show this to lead to the strange "sticky string" behaviour, where the part of the string in collision with the fretboard does not come back from the fretboard though its surrounding parts pull it up.

They also show that, when we account for this constant offset difference, the +ve and -ve waveguides are no more discontinuous, thereby making the string motion more physically correct.

In the following sections, the implementation that was done as part of this project is detailed along with results.

III. IMPLEMENTATION

A. Rank and Kubin method

The Rank and Kubin method [1] is implemented as shown in Fig. 2. The waveguides are divided into sections, with one section corresponding to the free length of the string (not limited by the fretboard) and the other sections corresponding to the ones along the fretboard length. The lengths of the delay lines are chosen according to the desired pitch of the note, as $N = \lfloor f_s/(2.f_0) \rfloor$, where N is the length of each of the +ve and -ve going waveguides and f_0 is the desired pitch.

In this implementation, n_is for $i = 2, 3, ..., k$, the lengths of the delay line sections between limiting blocks, are chosen to be equal to 4. n_1 is chosen to be equal to $0.25 * N$, as the number of fretboard delay sections are chosen to be in the ratio 3:1 with the number of free length delays (assuming the fretboard extends over $3/4^{th}$ the length of the string from the head-end to the body-end). k, the number of delay line sections is thus computed as $k = 0.75(N/4) + 1$. Each limiting block is implemented as shown in the right part of Fig. 1.

The reflection filter at the body-end is simply a sign inversion as a perfectly rigid termination is assumed. At the head-end, a moving average filter with transfer function $R_R(z) = 0.45(1+z^{-1})$ is chosen as it is found to dampen the output to produce a perceptually appealing slapbass pluck.

B. Krishnaswamy and Smith method

The implementation in this case [5] is essentially the same as the Rank and Kubin method [1], except that in the case of a collision we add a constant offset to either one of the right or left waveguide sections to maintain the continuity of the waveguides.

IV. RESULTS

A. Rank and Kubin method

1) Plucked case: For the "plucked" case, a triangular initial displacement as shown in Fig. 3 is given as input.

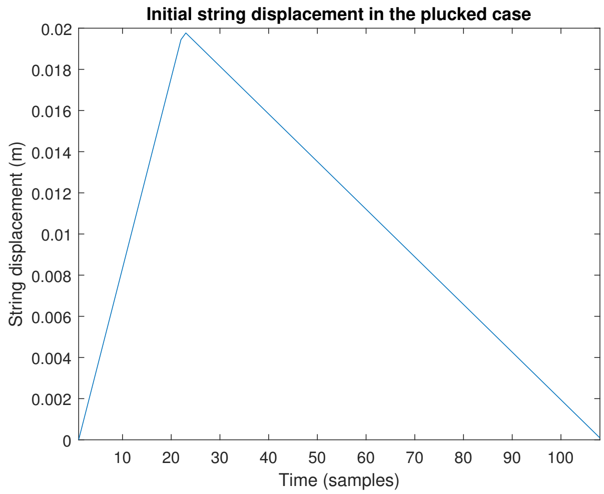

Fig. 3. Initial string pluck input displacement

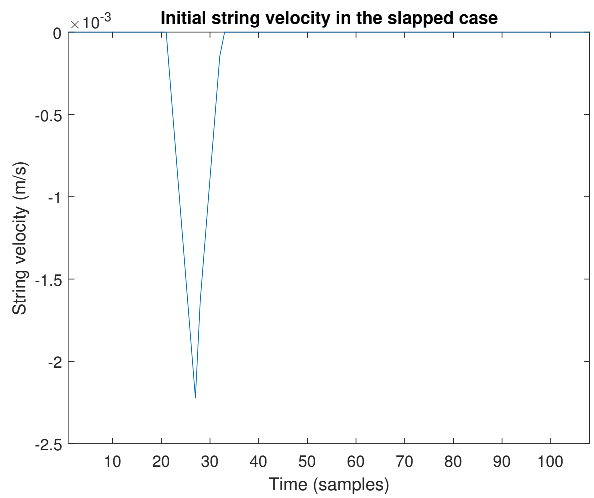

Fig. 4. Initial string velocity input in the slapped case

2) Slapped case: For the ”slapped” case, an initial velocity pulse as shown in Fig. 4 is taken as input. The initial displacements of each of the +ve and -ve going waveguides are computed by setting the sum of the waveguides as zero and the difference between them to the cumulative sum of the velocity pulse till that point.

The result of running the simulation is shown in Fig. 5 for the plucked case. As can be seen, the string displacement is limited by the fretboard. One can also observe that the string has a tendency to stick to the fretboard and not rise up. This confirms the observation made by the authors in [5].

The simulation result for the slapped case is shown in Fig. 6. As can be seen, the initialization is not right, leading to the end points of the string not having a displacement of zero. The implementation needs to be debugged to resolve this issue.

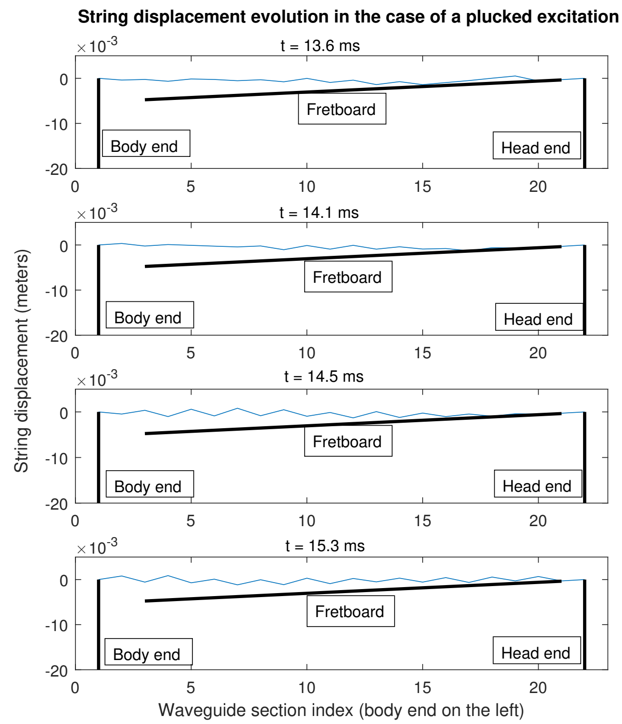

Fig. 5. String displacement evolution in the plucked case - the ”sticky string” behaviour can be clearly seen

B. Krishnaswamy and Smith method

An attempt was made to compensate for mismatched offsets being introduced by the Rank and Kubin method, as proposed in [5]. But, the output is observed to blow up after a certain duration of time, as seen in Fig. 7. This is most likely because the case of multiple collisions occurring at the same time is not handled well in the implementation. In the case of multiple simultaneous collisions, the waveguide needs to be broken into multiple sections. If we compensate for the offsets in one section for one collision and then compensate the offsets in another section for another collision, we may still end up having mismatched waveguides. The understanding of the paper itself needs to be revisited to correct this.

V. CHALLENGES FACED

The challenges faced during the course of the project are as follows:

- Though the papers [1], [5] looked simple, understanding the concept was a task in itself. A few iterative readings had to be done to get the concepts right
- Even with a fairly clear understanding of the papers, the Krishnaswamy and Smith paper [5] proved to be a tough one to implement, because there are some implementation details which are not apparent from the paper. An example of this would be the way to handle multiple simultaneous collisions, in which case a wrong implementation would lead to unrealistic behaviour, such as an unstable response in this case.

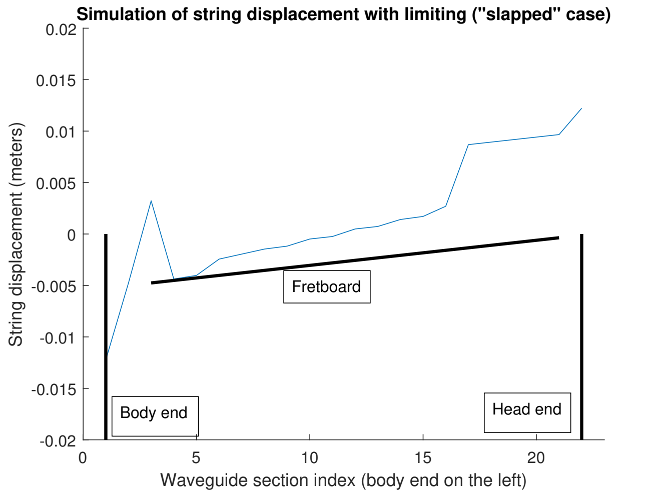

Fig. 6. Snapshot of string displacement for the "slapped" case - due to wrong initialization, the ends of the strings are not at zero

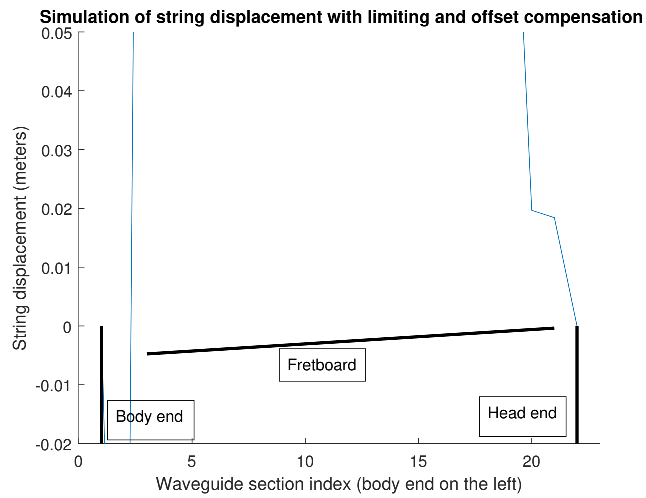

Fig. 7. Snapshot of string displacement with the Krishnaswamy and Smith method - the displacement is blowing up because of an implementation bug

- In general, though digital waveguides appear simple, slight errors and even small implementation details which can be overlooked affect the end result drastically, which is why attention must be given to every detail.

VI. OPEN-ENDED QUESTION

There is a detail which is intriguing in digital waveguide-based modeling. For a given f_s, if a lower f_0 is desired, the only way to achieve it seems to be by increasing N. Even in [], the authors increase N to obtain a lower f_0. Consider the case where we have a desired f_0 of 50 Hz. If we choose $f_s = 44.1\ kHz$ we get $N = f_s/(2f_0) = 441$. However, because each spatial sample corresponds to a length of $\approx 7.8\ mm$, this would mean a string length of $441 * 7.8\ mm = 3.44\ m$! A typical guitar string would have a length of 0.5 m up to 0.6 m, thus making our digital waveguide to be physically unrealistically long.

In a real string, the thickness of the string greatly influences f_0, with thicker strings having lower f_0. But in the digital waveguide formulation, this seems to stem from the fact that c is considered to be a constant, whereas it is in fact dependent on K and ϵ. How can one account for the string tension and thickness in the digital waveguide world seems to be an open-ended question.

VII. CONCLUSION AND FUTURE WORK

An attempt was made to implement two papers [1], [5] on slapbass modeling using digital waveguides. Results show that the Rank and Kubin method [1] was successfully implemented, barring the exception of the "slapped" case. The Krishnaswamy and Smith method [5] needs to be revisited in order to get a clearer detail. As shown in a demo during the in-class presentation, the plucked slapbass sound is perceptually similar to the actual sound. Implementation was done using Matlab. A script which randomizes different parameters like duration, pluck amplitude and $f0$ was also implemented and presented in class.

Future work would be to debug the failure cases and correct them. An interesting extension of this would be to use similar methods to model sound production in a Tanpura (Tamboora), which is a drone instrument used in Indian classical music. In the case of the Tanpura, the geometry is different from the electric guitar and collisions are restricted to a short length of the string in the region of a flat bridge. Nevertheless, since the Tanpura also involves collisions of the string limiting its displacement, the methods used here seem to be promising, and a suitable adaptation might lead to interesting results.

REFERENCES

[1] E. Rank and G. Kubin, "A waveguide model for slapbass synthesis," 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, Munich, 1997, pp. 443-446 vol.1. doi: 10.1109/ICASSP.1997.599670

[2] J. O. Smith. "Physical Modeling using Digital Waveguides", Computer Music Journal, special issue on Physical Modeling of Musical Instruments, Part I, Volume 16, no. 4, pp. 74 - 91, Winter, 1992

[3] K. Karplus A. Strong . "Digital Synthesis of Plucked-String and Drum Timbres". Computer Music Journal. 7. 43-55, 1983 doi:10.2307/3680062.

[4] D. A. Jaffe, J. O. Smith. "Extensions of the Karplus-Strong plucked-string algorithm", Computer Music J., Vol. 7, No. 2. (1983), pp. 56-69

[5] A. Krishnaswamy and J. O. Smith, "Methods for simulating string collisions with rigid spatial obstacles," 2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (IEEE Cat. No.03TH8684), New Paltz, NY, USA, 2003, pp. 233-236. doi: 10.1109/ASPAA.2003.1285874