MODELING METHODS FOR THE HIGHLY DISPERSIVE SLINKY SPRING: A NOVEL MUSICAL TOY

Julian Parker, Henri Penttinen
Department of Signal Processing and Acoustics,
Aalto University,
Espoo, Finland
Firstname.Lastname@tkk.fi

Stefan Bilbao
Acoustics and Fluid Dynamics Group/Music,
University of Edinburgh,
Edinburgh, UK
Stefan.Bilbao@ed.ac.uk

Jonathan S. Abel
CCRMA, Dept. of Music
Stanford University
Palo Alto, USA
abel@ccrma.stanford.edu

ABSTRACT

The 'Slinky' spring is a popular and beloved toy for many children. Like its smaller relatives, used in spring reverberation units, it can produce interesting sonic behaviors. We explore the behavior of the 'Slinky' spring via measurement, and discover that its sonic characteristics are notably different to those of smaller springs. We discuss methods of modeling the behavior of a Slinky via the use of finite-difference techniques and digital waveguides. We then apply these models in different structures to build a number of interesting tools for computer-based music production.

1. INTRODUCTION

The 'Slinky' is a child's toy, which consists of a large helical spring. It was invented by Richard James in the early 1940s [1] and it is notable for its ability to 'automatically' walk down stairs after being set in motion by a small initial push. Acousticians (e.g. Matti Karjalainen) use the Slinky as a tool to explain and exemplify transversal and longitudinal wave vibrations. This paper treats the Slinky as a sounding object that can be digitally modeled and used as a musical tool. The initial idea arose from the observation that the Slinky makes laser gun-like sounds. This sound is audible when the one end of the Slinky is placed by the ear, while the other end is let hang freely, and the edge of the helix is tapped, e.g., with a finger.

In Section 2 of this paper, we present measurement results of a 'classic' metal Slinky, and draw conclusions about its behavior relative to smaller springs. In Section 3, we propose a continuous model for the vibration of the Slinky, and from this continuous model develop discrete models utilizing finite-difference and digital waveguide techniques. In Section 4, we propose two signal processing structures which allow the modeled Slinky to be used as a musical tool and audio effect. Supplementary materials, including audio examples and audio-processing plug-ins are available at the website associated with this paper[1].

[1] http://www.acoustics.hut.fi/go/dafx10-slinky

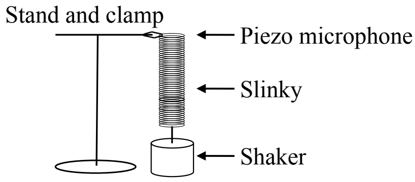

Figure 1: Measurement setup.

2. MEASUREMENTS

The Slinky was arranged as shown in Fig. 1, driven longitudinally and detected via a piezo-electric transducer. The Slinky was assumed to be linear and time-invariant and its impulse response measured using a sine-sweep method [2]. The measured thickness of the ribbon forming the Slinky coil varied in the range from 0.076 cm to 0.089 cm, but typically was around 0.084 cm. The width of the Slinky ribbon was measured to be 0.24 cm and inner coil diameter was 6.0 cm. The Slinky had 75 turns.

A prepared Slinky was stretched so that adjacent coils did not touch each other and placed in the vertical direction between a holder and a shaker (see Fig. 1). A plastic disc was glued to each end of the measured Slinky. In addition, a piezoelectric microphone was attached to the opposite end. The other plastic disk was attached to the shaker on the ground, and the end with the piezo-element was attached to a stand with a clamp at a height of 2 m above the ground. A 21 s long logarithmic sweep from 20 Hz to $f_s/2$ was used as an excitation. A sampling frequency of $f_s = 44.1$ kHz was used.

Figure 2 shows the spectrogram of the impulse response of the Slinky, derived from the sine-sweep measurement. As can be seen in the spectrogram, the impulse response consists of a repeating series of increasingly dispersed echoes, with low frequencies traveling more slowly than high frequencies. The form of the response looks somewhat different to that of the smaller springs used in spring reverberation units, as it appears to lack the primary set of dispersive echoes exhibited by smaller springs in the region below 3-4 kHz [3]. This result is consistent with the model presented

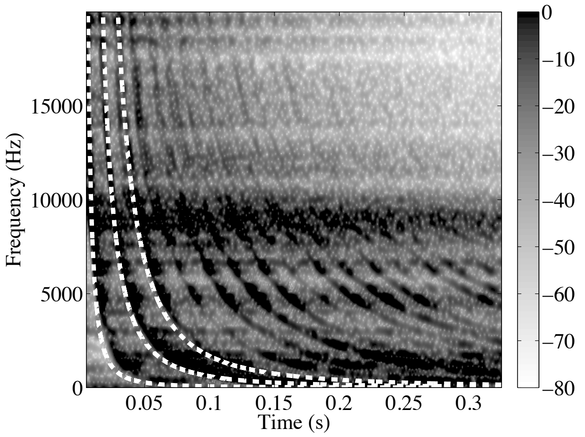

Figure 2: Spectrogram of a measured Slinky response.

in [3], which predicts that for the typical measurements of a Slinky spring, this second set of dispersive echoes would be present only at very low frequencies and hence be both inaudible and not visible in the spectrogram. Figure 2 also includes three white dashed curves representing three echoes produced by a continuous dispersion model discussed in Sec. 3.

3. MODELING METHODS

Mathematical modeling of helical spring vibration is a relatively mature topic, although its consideration from the perspective of audio frequency behavior is newer [4, 3, 5, 6]. The measurements given in Sec. 2 show that a significant part of the behavior of smaller springs is absent in the case of the larger Slinky spring. Therefore, a model which reproduces these behaviors is not necessary. The behavior of the Slinky is more reminiscent of the results produced by models of a bar [4, 5] or stiff string, and indeed early models of spring vibration and buckling treated the spring as a uniform bar [7]. Therefore, a reasonable starting point for modeling the behavior of a Slinky would be the Euler-Bernoulli ideal bar equation, given in dimensionless form:

$$\frac{\partial^2 u}{\partial t^2} = -\kappa^2 \frac{\partial^4 u}{\partial x^4} + \left[-2\sigma_o \frac{\partial u}{\partial t} + 2\sigma_1 \frac{\partial^3 u}{\partial t \partial x^2} \right] \tag{1}$$

where u represents transverse displacement, x is a coordinate running along the bar, t is time and κ is a dimensionless constant which encompasses scale, stiffness and material properties. The terms in brackets are additions to the ideal bar equation which represent loss in the system. The parameters σ_0 and σ_1 control the loss characteristics [8]. For a real bar, κ can be specified exactly in terms of values such as material density, Young's modulus and length. However, when modelling a Slinky as a bar these values become abstract and difficult to measure effectively. We therefore instead treat κ as a free parameter broadly effecting the dispersive behavior of the system, which can be adjusted to fit measured results (or by ear, for artistic purposes).

We assume that an impulse is being transmitted into the system at $x = 0$, and received by a transducer at $x = 1$. If we examine the dispersion relation of the lossless version of the system ($\sigma_0, \sigma_1 = 0$), we can derive an expression giving the time taken to perform one end-to-end traversal of the system at a particular frequency:

$$T_D = \frac{1}{2\sqrt{2\pi\kappa f}} \tag{2}$$

This expression gives the shape of the first dispersive echo to arrive in the impulse response. We can then use this expression to estimate a reasonable value of κ for the measured Slinky response. This was achieved by filtering the measured signal with a narrow FIR band-pass filter centred on a certain frequency. The distance between the major peaks in the filtered time-series gives the time taken for two traversals of the Slinky at that frequency. With that information and (2), we can estimate a value of κ. This process was repeated at a number of frequencies, and the mean of the κ estimates taken. This resulted in the value $\kappa = 0.06$. In Fig. 2 the three first echoes are plotted as dashes white curves when $\kappa = 0.06$ on top of the measured response. Agreement with the measured results appears to be reasonably close.

With this continuous model in place, we can approach the problem of building a discrete model via a number of techniques. Here, we examine (i) direct discretization of Equation 1 with a finite-difference (FD) technique, and (ii) approximation of the response with a modified single-delay loop (SDL) [9] digital waveguide (DWG) [10] model, similar that of [6].

3.1. Finite-Difference Model

Discretization of differential equations via the application of finite-difference techniques is a mature topic in many fields of science and engineering, but audio applications of the technique have only recently been explored [8]. A flexible and conceptually simple method of constructing a finite difference scheme is by the application of difference operators, which are discrete approximations to differential operators. These difference operators are applied to a number (one in this application) of 'grid functions', which are discrete version of the dependent variables of the system. For a system in one spatial dimension and time, the grid function is a 2D array of values. Each 'row' of such a grid function represents the distributed state of the system at a particular discrete time-step. System 1 is second order with respect to time, and therefore in this case the grid function need only contain two rows – representing the two previous time steps that are necessary to calculate the new state of the system. Grid functions can be considered to be analogous to the state-variables of a system.

It is important to note that there are many discrete approximations to a derivative operator, corresponding to different forms of numerical integration – Forward Euler, Backwards Euler, Runge-Kutta etc. These different forms of operator can be mixed and matched to produce many discretizations of a continuous system with different properties with respect to accuracy, stability and computability. In this case, the operators were chosen as follows:

$$\delta_{t+}\delta_{t-}u = -\kappa^2\delta_{x+}\delta_{x+}\delta_{x-}\delta_{x-}u - 2\sigma_0\delta_{t.}u + 2\sigma_1\delta_{x+}\delta_{x-}\delta_{t-}u \tag{3}$$

where u now refers to the discrete grid function of the system, δ denotes a difference operator, and its subscript denotes its type. The letter of the subscript denotes the variable against which the differentiation occurs, and the symbol following the letter denotes the method of integration. For example, δ_{t+} denotes Forward-Euler integration of time, δ_{t-} denotes Backwards-Euler integration and $\delta_{t.}$ represents Crank-Nicolson integration. In this case the time difference operators were chosen in order to produce an *explicit* discrete model, and the spatial difference operators were chosen so that the update of point on the grid function depends on a distribution of points centred on the update point. This formulation of the equation may now be expanded out to provide a

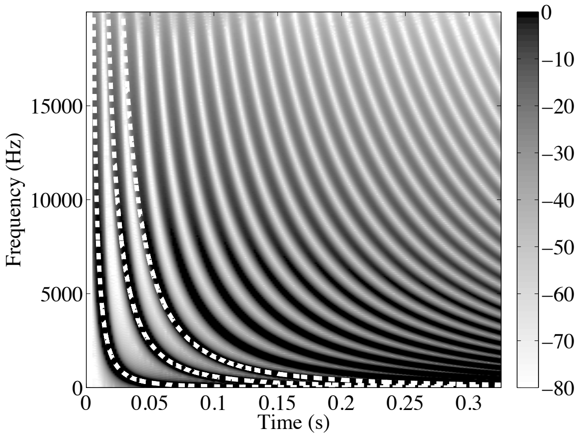

Figure 3: Spectrogram of a finite-difference model.

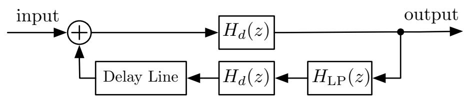

Figure 4: Block diagram of SDL waveguide model.

scheme for updating the grid function at every time-step. Pivoting boundary conditions $\left(u = \partial^2 u/\partial x^2 = 0\right)$, are chosen for the continuous model and discretized similarly. The resulting scheme shows good numerical properties, although as is typical for this method of discretization, accuracy degrades at high frequencies – producing some artificial high-frequency dispersion. This effect can be eliminated by oversampling.

Figure 3 shows the spectrogram of the impulse response of the output of the finite difference scheme when excited by an impulse. We choose $\kappa = 0.06$, and the values of σ_0 and σ_1 are set to produce a gentle roll-off of reverberation time as frequency increases. The results appear to be in reasonably close agreement with the measured properties of the Slinky, at least in terms of dispersive behavior. The presence of several peaks in the reverberation time of the real Slinky is not reproduced, but this quality is likely a function of the material of the real Slinky which is modeled only crudely in this scheme. Sonically, the result of this model is close to the recording of a real Slinky. It lacks a certain diffuse, reverberant quality, but the basic character of the sound is accurately reproduced.

3.2. Digital Waveguide Model

The Slinky can be seen as a very stiff string. For this approach, waveguide modeling [10] is a good starting point. Moreover, for efficiency reasons we will apply an SDL [9, 6] version and need only one delay line. The structure of the modified single-delay loop digital waveguide (SDL DWG) model for the Slinky is shown in Fig. 4 and explained below. The Slinky does not a have a very easily defined fundamental frequency. Hence, we use the continuous model and (2) for deriving the length of the delay line and dispersive behaviour of the model.

The dispersive nature of the Slinky is modeled with a chain of allpass sections using the method presented in [11]. This method approximates a given group delay with a chain of second-order all-

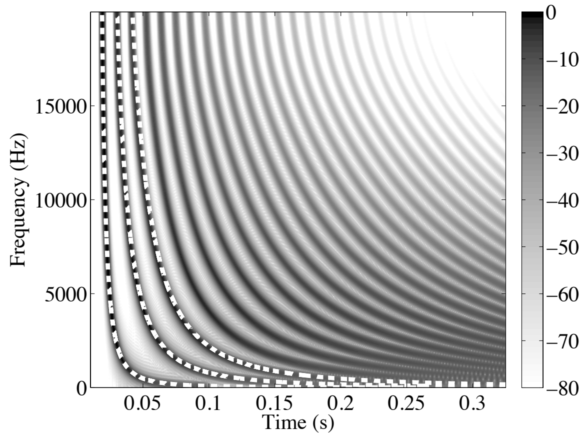

Figure 5: Spectrogram of an SDL waveguide model.

pass filters. At low frequencies (when f approaches DC) the group delay given by (2) approaches infinity. Hence, frequencies below 100 Hz were restricted to the value obtained at 100 Hz (when $\kappa = 0.06$ the group delay at this frequency is 3350 samples). With these values the result was a chain of 113 biquads. This allpass chain $H_d(z)$ models the first dispersive echo shown in Fig. 2. Therefore, one chain $H_d(z)$ is placed in the direct path. Moreover, one $H_d(z)$ chain is placed in the feedback path to simulate the traveling of the impulse back to the input end of the Slinky.

The density of the dispersive echos is calculated as the time delay between two consecutive echos with (2) when f= 22 kHz. This is because at high frequencies the repeating echos have a relatively constant time difference at neighboring frequencies, compared to time differences of echos at low frequencies. When rounding to integers this gives us a delay line length of 484 samples. The length of the delay line has to be compensated by the group delay of two $H_d(z)$ chains at 22 kHz, i.e., the delay line length is shortened by the delay caused by the model in the feedback loop at 22 kHz. A one-pole filter discussed in [12] was used for modeling the frequency-dependent decay $H_{\rm LP}(z)$. Fig. 5 shows the spectrogram of the waveguide model. Again, the echos at high frequencies appear as in measurements and the dispersive behavior is matched nicely.

4. APPLICATIONS

The motivation for the models described in Sec. 3 was the idea of using the Slinky as a tool for producing interesting musical sounds and effects. To this end, we implemented the Slinky models described above as objects in Cycling74's Max/MSP [13] programming environment. These objects were then used to produce larger audio-effect and instrument structures, implemented in Max/MSP and 'Max For Live' as plugins for the Ableton Live music production environment [14]. The objects and plugins, along with further information on their structure, are available at `http://www.acoustics.hut.fi/go/dafx10-slinky`.

4.1. Feedback Slinky Network

The *Feedback Slinky Network* (FSN) is inspired by the idea of the Feedback Delay Network (FDN), as introduced by Jot [15]. We implement the FSN as four Slinky models connected by a matrix specifying the gains between the outputs and inputs of each of the models. The κ parameter and loss characteristics of each

Slinky can be varied by the user. This structure behaves notably differently to the FDN, as unlike the delay-line elements of the FDN the Slinky model elements of the FSN have an extended impulse response. Consequently, feedback between these elements can quickly cause unbounded growth in the system. To counteract this effect, we place $\tanh$ wave-shaping elements after each Slinky model, which limit the maximum signal value possible.

A feedback matrix containing low values, combined with low values of κ, produces a reverb-like effect. Raising the value of κ for the Slinky models results in a structure that sounds more like a complex resonator. Raising the values in the feedback matrix results in self-oscillation of the system, turning it into more of a sound-source than an processing device. Modulating the κ values of the Slinky models whilst the system is self-oscillating produces interesting shifting inharmonic drone sounds.

4.2. Highly Dispersive String-Instrument

As discussed above, in Sec. 3.2, the behavior of the Slinky strongly resembles that of an extremely dispersive string. We can therefore apply a single Slinky model, almost directly, to produce a modeled instrument that behaves like an extremely dispersive version of a string instrument. We implement such an instrument as a single Slinky model, with a user controllable excitation method. The excitation method consists of a filtered noise-source, combined with an amplitude envelope. The noise-filter consists of independently variable one-pole high-pass and low-pass filters connected in series. By manipulation of the amplitude envelope and noise-filter parameters, the user can excite the model in a variety of ways – ranging from short pluck-like excitations to slow excitations reminiscent of bowing [10]. Reception of a MIDI note-on message triggers the amplitude envelope of the excitation signal. Variation of pitch can be accomplished by altering the value of κ, in the case of the finite difference model, or by altering the delay-line length in the case of the modified SDL waveguide model. Classification of the perceived pitch of the model is difficult due to the inharmonic qualities, therefore no attempt is made to tune the instrument exactly. Instead, the user can specify the way in which the 'pitch' scales with the incoming MIDI note number. The resulting instrument is capable of producing a variety of sounds, from sci-fi laser-gun zaps to more conventional string-like tones.

5. CONCLUSIONS

This paper presented sonic observations, digital models, and audio applications of the well-known spring toy called the Slinky. The main acoustic observation is that the helical spring of the Slinky is highly dispersive. The measured impulse response consists of decaying echos that have a dispersive character. Based on this analysis, we proposed a continuous model of Slinky vibration. This model was then used to produce discrete models via finite-difference and digital waveguide techniques. Both models recreate the basic characteristics of the Slinky response fairly well. The models were then developed into two parametric musical devices or sound effects. The Feedback Slinky Network consists of Slinky models that are connected through a feedback matrix. This network can create filtering effects from reverb-like sounds to a self-oscillating system. Another application uses the Slinky to construct a model of a highly dispersive and inharmonic string-like instrument, which is played by excitation with a filtered and shaped noise signal. In both applications the user can control the dispersiveness and decay characteristics of the models. The results of these applications are interesting, and not easily reproduced using physical Slinky springs.

6. ACKNOWLEDGMENTS

This work has been financed by the Academy of Finland (projects no. 128689 and 122815). The authors would like to thank Vesa Välimäki,Veronique Larcher, Stephen Backer and Chris Warren for their assistance at various points during this project.

7. REFERENCES

[1] "Slinky," URL: http://en.wikipedia.org/wiki/Slinky, 2010.

[2] A. Farina, "Simultaneous measurement of impulse response and distortion with a swept-sine technique," in *AES 122nd Convention, Vienna*, 2000.

[3] J. Parker and S. Bilbao, "Spring reverberation: A physical perspective," in *Proceedings of the 12th International Conference on Digital Audio Effects (DAFx09), Como*, 2009, pp. 416 – 421.

[4] J.D. Parker, "Spring reverberation: A finite difference approach," M.S. thesis, University of Edinburgh, 2008.

[5] S. Bilbao and J. Parker, "A virtual model of spring reverberation," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 4, pp. 799–808, May 2010.

[6] J.S. Abel, D.P. Berners, S. Costello, and J.O. Smith III, "Spring reverb emulation using dispersive allpass filters in a waveguide structure," in *Proc. of the 121st Convention of the AES*, San Francisco, California, 2006.

[7] J.A. Haringx, *On Highly Compressible Helical Springs and Rubber Rods, and Their Application for Vibration-Free Mountings*, Philips Research Laboratories, 1950.

[8] S. Bilbao, *Numerical Sound Synthesis*, John Wiley and Sons, 2009.

[9] M. Karjalainen, V. Välimäki, and T. Tolonen, "Plucked-string models: from the Karplus-Strong algorithm to digital waveguides and beyond," *Computer Music Journal*, vol. 22, no. 3, pp. 17–32, 1998.

[10] J. O. Smith, "Physical modeling using digital waveguides," *Computer Music Journal*, vol. 16, no. 4, pp. 74–91, 1992.

[11] Jonathan S. Abel and Julius O. Smith, "Robust design of very high-order allpass dispersion filters," in *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06)*, Montreal, Quebec, Canada, Sept. 18–20, 2006, pp. 13–18.

[12] V. Välimäki, J. Huopaniemi, M. Karjalainen, and Z. Jánozy, "Physical modeling of plucked string instruments with application to real-time sound synthesis," *J. Audio Eng. Soc.*, vol. 44, pp. 331–353, 1996.

[13] "Cycling 74," URL: http://www.cycling74.com/, April 2010.

[14] "Ableton," URL: http://www.ableton.com/, April 2010.

[15] J.M. Jot and A. Chaigne, "Digital delay networks for designing artificial reverberators," in *AES 90th Convention, Paris*, 1991.