# Parametric Spring Reverberation with Automatic Parameter Calibration

MUMT-618 Final Project Report

Eto Sun
(eto.sun@mail.mcgill.ca)

## 1 Introduction

Digitalizing analog electronic effects, also known as "virtual analog," has gained great attention in recent years [1]. The use of helical springs for artificial reverberation was a cost-effective method in the past. Due to its unique sound characteristics, the device is still used in modern music production. For this reason, modeling the spring reverberation in the digital context is necessary.

It is difficult to model a spring due to its longitudinal and transverse wave propagation and highly dispersive behavior [2]. There are two primary types of approaches used to simulate spring reverberation. The first type involves analyzing the impulse response and spectrogram of the spring reverb from a perceptual perspective and then reproducing it in a parametric way. For example, since the time-frequency representation of the response of a single spring contains two sequences of chirps, Välimäki *et al.* utilize a cascade of all-pass filters to synthesize the chirp signal and modulated delaylines with a feedback loop to reproduce multiple pulses [3]. The second category involves modeling and discretizing the vibration of a helical spring from a physical perspective. Bilbao and Parker, for instance, selected a simple physical model of the vibration of a helical structure and implemented the finite difference schemes in the time domain for the discretization [2].

In this project I aim to implement the parametric spring reverb model from [3], construct the automatic parameter calibration for the reverb model from [4], and further improve the automation process. All models are implemented in Matlab.

This project report is organized as follows: in the first section, I describe the details for implementing the parametric spring reverb model in [3]. In the second section, I present the implementation of the automatic calibration process in [4]. Some modifications are proposed to improve the estimation of the parameters. The results of the model with automatic calibration and conclusions are discussed in the third section. The last section concludes this report.

## 2 Parametric Spring Reverb Model

By analyzing the impulse response of a spring unit from a reverberation tank, we can see from figure 1 that the response contains two sequences of decaying chirps. The low-frequency chirps are stronger and disappeared at around 4 kHz, refers to the *transition frequency* $f_{\text{tr}}$. The high-frequency chirps are much weaker and located in all frequencies. We will use cascaded filters and delay lines to produce these two chirp sequences separately.

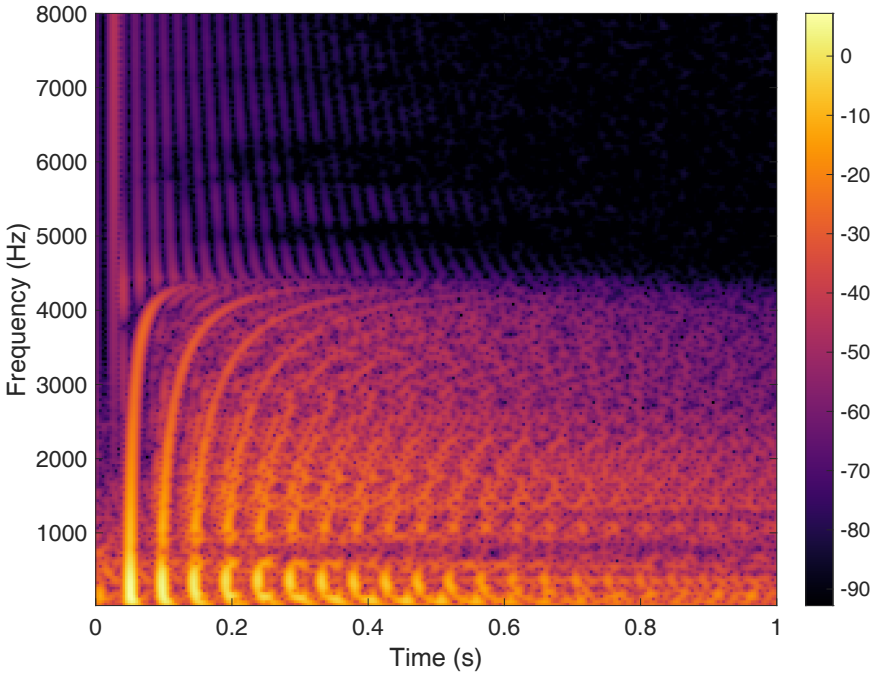


Figure 1: The spectrogram of a measured single spring response, obtained with discrete gabor transform (DGT) using the Gaussian window with a length of 512 samples. The magnitude is in dB scale.

## 2.1 Modeling the low-frequency chirp sequence

The first chirp of this sequence can be obtained from passing a unit impulse to the spectral delay filter [5]. The spectral delay filter is a cascade of many identical low-order all-pass filters. The transfer function of the all-pass filter is:

$$H_{\text{low}}(z) = \frac{a_1 + z^{-K}}{1 + a_1 z^{-K}} \tag{1}$$

The parameter $K$ determines the frequencies with maximum delay, and can be represented by the transition frequency $f_{\text{tr}}$ as $K = f_s/(2f_{\text{tr}})$, where $f_s$ is the sampling rate. Due to the fact that $K$ may be not an integer, a fractional delay should be incorporated. In this situation, the first-order all-pass filter for interpolation is used because of its simplicity. The transfer function with fractional delay is:

$$H_{\text{low}}(z) = \frac{a_1 + \frac{a_2+z^{-1}}{1+a_2 z^{-1}} z^{-K_1}}{1 + a_1 \frac{a_2+z^{-1}}{1+a_2 z^{-1}} z^{-K_1}} = \frac{a_1 + a_1 a_2 z^{-1} + a_2 z^{-K_1} + z^{-(K_1+1)}}{1 + a_2 z^{-1} + a_1 a_2 z^{-K_1} + a_1 z^{-(K_1+1)}} \tag{2}$$

The parameter $K_1 = \lfloor K - 1/2 \rfloor$ is one less than the integral part of $K$. The coefficient $a_2 = (1-d)/(1+d)$ determines the fractional delay, where $d = K - K_1$.

To cascade $M$ identical filters, I compute the output iteratively to avoid accuracy issue because if the cascaded filters are converted into a very high order filter using convolution, some coefficients will be very large or very small. I also create matrices of shape $(K_1 + 1) \times M$ to store the state values of the cascaded filters.

The generated chirp needs to be low-passed to remove the components above the transition frequency $f_{\text{tr}}$. I use a $10^{\text{th}}$-order lowpass elliptic filter with 1 dB of passband ripple, 60

dB of stopband attenuation, and a passband edge frequency of $0.95 f_{\text{tr}}/(f_s/2)$ Hz. The elliptic filter is implemented via the Matlab function `ellip`.

The chirp needs to be equalized for better approximating the spectral shape of the measured chirp, an IIR filter will be used with the following transfer function,

$$H_{\text{eq}}(z) = \frac{1-R^2}{2} \frac{1-z^{-2K_{\text{eq}}}}{1+a_{\text{eq1}}z^{-K_{\text{eq}}}+a_{\text{eq2}}z^{-2K_{\text{eq}}}} \tag{3}$$

where $R = 1-(\pi B K_{\text{eq}})/f_s$ is determined by peak frequency $f_{\text{peak}}$ and $-3$-dB bandwidth $B$. The coefficients of $a_{\text{eq1}}$ and $a_{\text{eq2}}$ are $-(1+R^2)\cos(2\pi f_{\text{peak}} K_{\text{eq}}/f_s)$ and $R^2$, correspondingly. The delay length $K_{\text{eq}}$ can be any integer less than $K$. In my implementation, $K_{\text{eq}} = \lfloor K \rfloor$.

The sequence of chirps can be produced by a feedback delay line. The delay time $T_d$ is mainly based on the estimation from recorded impulse response. In order to attenuate the feedback chirps, a gain factor $g_{\text{lf}} < 1$ should be used. The delay line can be further divided into three parts. The total length of these delay lines is $L = T_d f_s - KM(1-a_1)/(1+a_1)$. The first delay line aims to delay the chirp, and does not contain a feedforward path. The second one is to create faint "pre-echoes" preceding each delayed chirps, thus contains a feedforward path with gain $g_{\text{echo}}$. The length of it is $L_{\text{echo}} = L/5$. The third one is a comb filter implemented by a feedforward path with gain $g_{\text{ripple}}$, which makes some frequencies decay faster. The length is $L_{\text{ripple}} = 2KN_{\text{ripple}}$, where $N_{\text{ripple}}$ is the number of combs below the transition frequency. The total length of the delay lines are modulated by a noise generator for the diffusion. The noise has an approximately Gaussian distribution. I use hard clipping to restrict the noise signal lies within the interval [-1, 1]. The modulation depth is introduced by multiply the noise signal with the parameter $g_{\text{mod,low}}$.

Due to the fact that there is an increased delay at very low frequencies in the measured response, an DC blocking filter should be introduced in the feedback loop. The transfer function is:

$$H_{\text{dc}}(z) = \frac{1-a_{\text{dc}}}{2} \frac{1-z^{-1}}{1-a_{\text{dc}}z^{-1}} \tag{4}$$

The block diagram of the structure to produce low-frequency chirp sequence is in Figure 2.

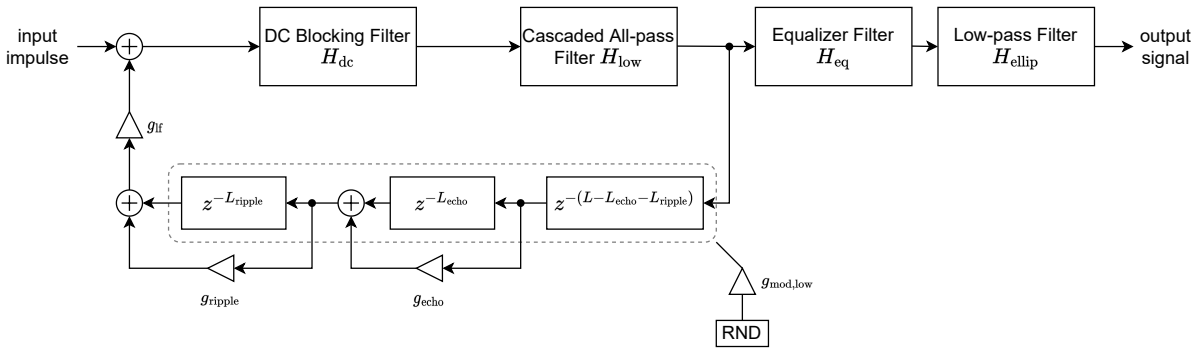


Figure 2: Block diagram of feedback structure used to generate the sequence of low-frequency chirps.

## 2.2 Modeling the high-frequency chirp sequence

The high-frequency chirp sequence has similar shape to the low-frequency one, therefore the techniques to used are also the same. The first chirp is produced via cascading all-pass

filters, with a negative filter coefficient $a_{\text{high}}$. The transfer function of one single all-pass filter is:

$$A_{\text{high}}(z) = \frac{a_{\text{high}} + z^{-1}}{1 + a_{\text{high}} z^{-1}} \tag{5}$$

The chirp sequence can be obtained by introducing a simple delay line. The delay line length is $L_{\text{high}} = L/2.3$ in my implementation. The length is also modulated by a noise generator with the modulation depth $g_{\text{mod,high}}$. The block diagram of this structure is in Figure 3.

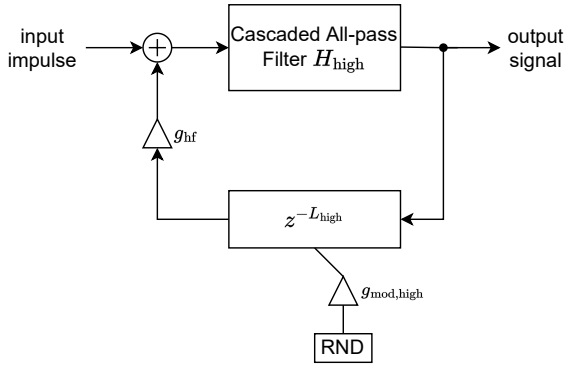


Figure 3: Block diagram of feedback structure used to generate the sequence of high-frequency chirps.

## 2.3 Combination of two components

The low and high frequency structures will be combined to form a complete spring model. The coupling coefficients $c_1$ and $c_2$ aim to add interaction between two structures. The block diagram of the structure is in Figure 4. To simulate a spring reverberation tank, several springs need be connected in parallel, as shown in Figure 5.

Components including various delay lines and filters are encapsulated into handle classes in Matlab. The single spring model is constructed as a high-level handle class that contains all components and their parameters.

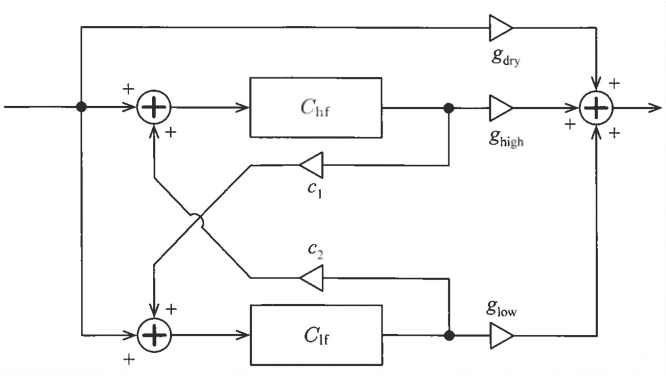


Figure 4: Block diagram of the total structure for modeling a single spring. The component $C_{\text{lf}}$ represents the low-frequency feedback structure in Figure 2, $C_{\text{hf}}$ is the high-frequency feedback structure in Figure 3.

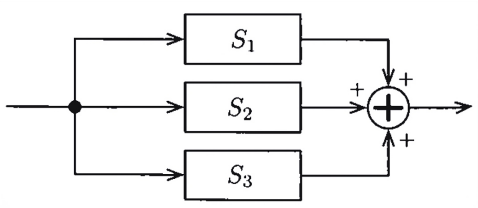


Figure 5: Block diagram of three spring models in parallel, where $S_1$, $S_2$, $S_3$ are the single spring components in Figure 4 with different parameters.

# 3 Automatic Parameter Calibration

The parametric spring reverb model includes many parameters, which makes digitizing the hardware spring reverberators challenging. Gamper *et al.* proposed a process to calibrate most of the parameters automatically from a measured spring impulse response [4]. However, the automatic process is not robust. For some measured impulse response, the algorithm cannot make a meaningful decision. This section will describe the original ideas from [4] and improve some algorithms further.

## 3.1 Global parameters

### 3.1.1 Chirp delay time

The chirp delay time $T_d$ is the distance between two low-frequency chirps. It can be estimated from the maximum absolute value of the autocorrelation (`xcorr` in Matlab) of the measured impulse response. The first 100 terms of autocorrelation will be set to zero to prevent they become the maximum value.

### 3.1.2 Transition frequency

The transition frequency $f_{\text{tr}}$ is the frequency boundary where the low-frequency chirps disappear. The original paper first obtains the time-frequency representation of the signal using short-time Fourier Transform (STFT) with Blackman window, 8192 frequency bins, and a time shift of 8 samples. Then a normalized autocorrelation of the STFT is calculated for each frequency bin, and the mean values of autocorrelation at each frequency bin will be evaluated. The maximum mean value should be at the transition frequency $f_{\text{tr}}$.

However, this algorithm may not be able to choose proper frequency because the impulse responses of some springs have high autocorrelation values at low frequencies, which causes the peak of the mean values appears at the low-frequency region. Therefore, I try to solve this problem by finding the last sudden drop of both energy and autocorrelation per frequency bin. I first represent the signal via the Equivalent Rectangular Bandwidth (ERB) filter bank with 512 filters for a better representation at low-frequency region. Then I calculate the pointwise multiplication of the following two curves: one is the mean value of the autocorrelation of the ERB filter bank for each frequency bin (same as the original paper, but the time-frequency representation changed), and the other is the 2-norm of the coefficients for each frequency bin. The transition frequency $f_{\text{tr}}$ can then be determined by the local minima with the highest frequency of the derivative (approximated with the first-order difference `diff`) of the multiplication curve. One example of a given measured signal is shown in Figure 6. The local minima can be detected via the Matlab function `findpeaks`.

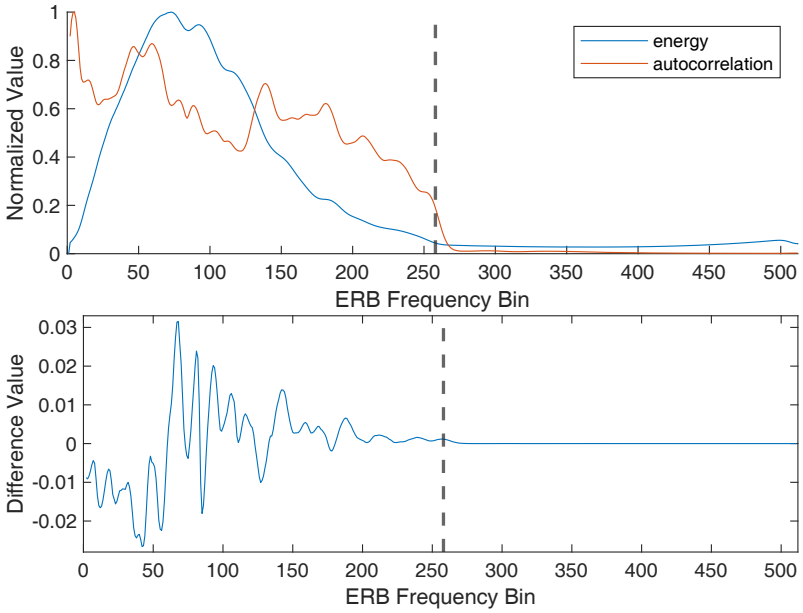


Figure 6: The top figure represents the 2-norm (energy) of the coefficients and the mean value of the autocorrelation for each frequency bin. The bottom figure represents the first-order difference of the multiplication of the energy and the autocorrelation curves, and the sign of the result is flipped. The transition frequency $f_{\text{tr}}$ is estimated at the last peak of the derivative of the multiplication of energy and autocorrelation curves, which is highlighted as the gray dashed line.

## 3.2 Low-frequency chirp parameters

### 3.2.1 Gain factor

The gain factor $g_{\text{lf}}$ can be estimated using the energy decay curve of the measured impulse response, which is computed via the backward integration of energy. The original paper chooses points with the interval $T_d$ and fits a line using the `polyfit` function in Matlab. Only the points greater than -10 dB of the energy decay curve will be involved in the fitting process. The decay per chirp is defined as the slope of the fitted line (converted to amplitude).

However, this method includes the energy of high-frequency chirps in the fitting process. In addition, the energy decay of low-frequency chirps is slowed down due to the interaction of different chirps. I use an IIR low-pass filter to remove the energy of high-frequency chirps. The cut-off frequency is the transition frequency $f_{\text{tr}}$. In order to make the chirp decay faster, I fit the line with the first three points. The comparison of these two methods is demonstrated in Figure 7.

### 3.2.2 Cascaded all-pass filter parameters

The original approach tries to extract all chirps one by one. In my implementation, I concentrate on extract the first chirp and fitting the all-pass filter iteratively. The parameters $a_1$ and $M$ need to be approximated according to the transfer function in Equation 2. The spectrogram of this signal is first calculated by the digital gabor transform (DGT), with Blackman window of length 512 samples, time shift of 8 samples, and 4096 channels. The spectrogram will be low-passed to remove any information above the transition frequency. For each frequency bin, the spectral peaks will be recorded using Matlab function `findpeaks`. A procedure is designed to find the continuous curve that capture the first chirp according to the spectral peaks. This procedure starts from the lowest frequency bin and time $T_d$. The curve value in a

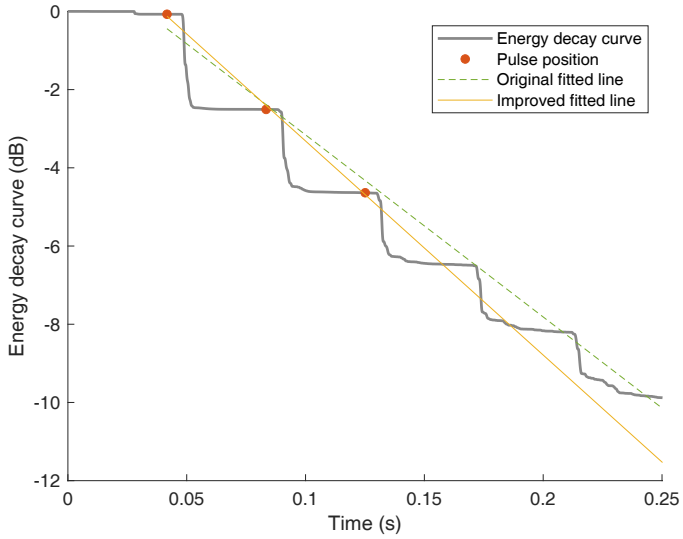


Figure 7: The energy decay curve and fitted line from the first three low-frequency pulse positions.

frequency bin is determined by the closest spectral peaks around the curve value in previous frequency bin. After constructing the curve, I use nonlinear least squares fitting using the Matlab function `lsqnonlin` to fitting the filter parameters iteratively. The objective function is to minimize the distance of spectral peak between synthesized chirp and measured chirp for each frequency bin. The initial values of $a_1$ and $M$ are 0.75 and 100, correspondingly.

#### 3.2.3 Chirp equalization filter parameters

The goal of the equalization filter is to change the spectral shape of a chirp. Therefore, we need to obtain the spectral distribution of first chirp, which can be approximated by calculating the Fourier transform of the impulse response from $T_d$ to $2T_d$. From the transfer function of the equalization filter in Equation 3, the delay length $K_{\text{eq}}$, peak frequency $f_{\text{peak}}$, and $-3$-dB bandwidth $B$ should be estimated. The fitting technique is the same as previous section. The goal is to make the chirp's spectral shape (after normalization) and the difference between the filter's frequency response for each frequency bin as small as possible. The initial values are $K_{\text{eq}} = 2$, $f_{\text{peak}} = 120$, and $B = 100$.

### 3.3 High-frequency chirp parameters

#### 3.3.1 Gain factor

Instead of scaling the gain factor of low-frequency chirp $g_{\text{lf}}$, I use the similar method to estimate $g_{\text{hf}}$ as for $g_{\text{lf}}$, except that the signal is high-passed.

The time interval between fit points is estimated from the maximum absolute value of the autocorrelation of the high-passed impulse response. The first 128 terms will be set to zero to avoid the procedure choose the initial unit impulse as the maximum.

#### 3.3.2 Cascaded all-pass filter parameters

The high-frequency chirps are also modeled using the digital gabor transform and the nonlinear least squares fitting as in modeling low-frequency chirps. The Blackman window length is set to 256 and total frequency bin is set to 2048 because each high-frequency chirp has less delay than low-frequency chirp, a higher temporal resolution is required to capture the delay. The procedure for finding the spectral peak of each frequency bin will start from

the high frequency bin. The start time will be the mean of locations of the spectral peaks minus its standard deviation. The objective function is the same as the low-frequency one. The initial values are $a_{\text{high}} = -0.5$ and $M = 200$.

# 4 Results

I test the calibration process with two measured springs. The measured impulse response signals were obtained from the companion website of [6]. The first impulse response is from a spring reverberation unit called Olson X-82. This spring has shorter helix length and lower sound. The second impulse response is measured from a Leem Pro KA-1210 guitar amplifier, which is larger than the previous one, and has brighter timbre. The physical attributes of these two springs are in Table 1.

The automatic calibration procedure is performed for the measured impulse response of these two springs, respectively. The calibrated parameters are shown in Table 2.

| **Physical Attributes** | **Olson X-82 Spring 2** | **Leem Pro KA-1210 Spring 1** |
|---|---|---|
| Helix Length (cm) | 6.5 | 16.3 |
| Helix Diameter (cm) | 0.61 | 0.44 |
| Number of Turns | 133 | 303 |
| Wire Diameter (cm) | 0.035 | 0.035 |
| Magnet Length (cm) | 0.5 | 0.4 |
| Magnet Diameter (cm) | 0.21 | 0.15 |

Table 1: The physical attributes of the two measured springs.

| **Spring Reverb Parameters** | **Olson X-82 Spring 2** | **Leem Pro KA-1210 Spring 1** |
|---|---|---|
| Chirp Delay Time ($T_d$) | 0.0417 | 0.0466 |
| Transition Frequency ($f_{\text{tr}}$) | 2060 | 4333 |
| Low Chirp Gain Factor ($g_{\text{lf}}$) | -0.80 | -0.64 |
| Low Chirp Cascade Number ($M_{\text{low}}$) | 72 | 200 |
| Low Chirp Filter Coefficient ($a_1$) | 0.4175 | 0.4233 |
| Equalizer Peak Frequency ($f_{\text{peak}}$) | 108.89 | 93.44 |
| Equalizer $-3$-dB Bandwidth ($B$) | 90.41 | 247.29 |
| High Chirp Gain Factor ($g_{\text{hf}}$) | -0.90 | 0.70 |
| High Chirp Cascade Number ($M_{\text{high}}$) | 158 | 219 |
| High Chirp Filter Coefficient ($a_{\text{high}}$) | -0.4746 | -0.3383 |

Table 2: The automatic calibration results of the parameter spring reverberation for the two measured springs.

After obtaining the parameters, the parametric spring reverberation model that was previously built is used to produce the simulated impulse responses. For those parameters that are not covered in the calibration process, the default values in [3] will be used, such as the modulation depth of the delay lines and the coefficient of the DC blocking filter. The comparison of the impulse response of the parametric model with the original measured signal is shown in Figure 8 and Figure 9.

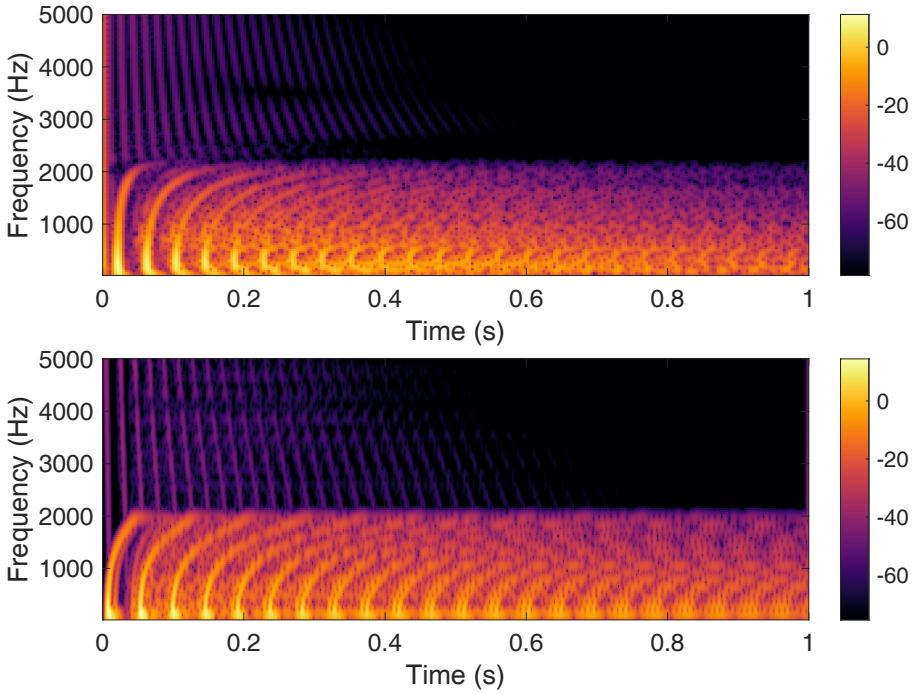


Figure 8: The spectrograms of measured impulse response and parametric model result. The top figure represents the spectrogram of the second spring of Olson X-82. The bottom figure is the output of the parametric spring reverb model with automatic parameter calibration.

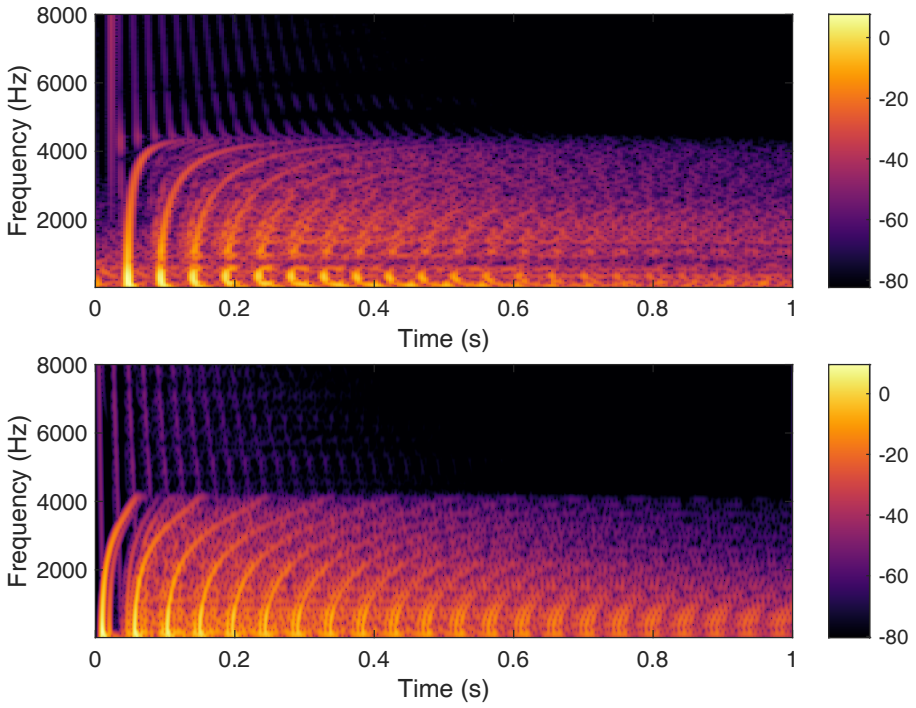


Figure 9: The spectrograms of measured impulse response and parametric model result. The top figure represents the spectrogram of the first spring of Leem Pro KA-1210. The bottom figure is the simulated impulse response from the parametric spring reverb model with automatic parameter calibration. The gain factors of chirps ($g_{\text{lf}}$ and $g_{\text{hf}}$) are scaled by 1.2 manually for a similar decay rate.

By comparing the spectrograms, we can observe that the transition frequency and delay time of low-frequency chirps are calibrated accurately. The spectral structure of the high-frequency chirps are mainly captured. However, the energy of the low-frequency chirp decays slower (in the case of Figure 8) or faster (Figure 9) than the measured signals. Manual scaling of the gain factors ($g_{\text{lf}}$ and $g_{\text{hf}}$) could be done for better results. The spectral shapes of the low-frequency chirps are different, especially near the transition frequency. The measured signal has longer group delay around the transition frequency. The difference is audible because the energy has discrepancies at around 1000 Hz and 4000 Hz. Besides, the details of the measured signal are not captured precisely such as the frequency of ripples (controlled by $N_{\text{ripple}}$), since these parameters are not calibrated automatically.

# 5 Conclusions

This project concentrates on implementing a parametric spring reverb model and an automatic parameter calibration procedure for the reverb model. Some optimization are proposed to improve the accuracy and stability of the calibration. The improved automatic calibration process is able to estimate most of the parameters for the spring reverb. Future work about this project could be optimizing the fitting process using state-of-the-art models, or preprocessing the input impulse response for more stable calibration. The automatic calibration of other parameters in the spring reverberation model could be investigated in the future.

# References

[1] V. Välimäki, S. Bilbao, J. O. Smith, J. S. Abel, J. Pakarinen, and D. Berners, "Virtual analog effects," in *DAFX: Digital Audio Effects*, John Wiley & Sons, Ltd, 2011, pp. 473–522. DOI: 10.1002/9781119991298.ch12.

[2] S. Bilbao and J. Parker, "A Virtual Model of Spring Reverberation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 4, pp. 799–808, May 2010. DOI: 10.1109/TASL.2009.2031506.

[3] V. Välimäki, J. Parker, and J. S. Abel, "Parametric spring reverberation effect," *Journal of the Audio Engineering Society*, vol. 58, no. 7/8, pp. 547–562, Jul. 2010.

[4] H. Gamper, J. Parker, and V. Välimäki, "Automated calibration of a parametric spring reverb model," in *Proceedings of the 14th International Conference on Digital Audio Effects (DAFx-11)*, Paris, France, Sep. 2011, pp. 37–44.

[5] V. Välimäki, J. S. Abel, and J. O. Smith, "Spectral Delay Filters," *Journal of the Audio Engineering Society*, vol. 57, no. 7/8, pp. 521–531, Jul. 2009.

[6] J. Parker and S. Bilbao, "Spring Reverberation: A Physical Perspective," in *Proceedings of the 12th International Conference on Digital Audio Effects (DAFx-09)*, Como, Italy, Sep. 2009.