

# WHAT ELSE SAYS “ACOUSTICAL CHARACTERIZATION SYSTEM” LIKE RON JEREMY?

*Andrew Greenwood*

Stanford University  
Center for Computer Research in Music and Acoustics (CCRMA)  
[Aeg165@ccrma.stanford.edu](mailto:Aeg165@ccrma.stanford.edu)

## ABSTRACT

An interface was created for the generation and basic analysis of acoustical test signals. The interface was created using Matlab’s GUIDE. The available test signals are linear and logarithmic sine sweeps, Golay pulse trains and an allpass smear. The available control parameters include Golay length, sweep time, the allpass filter coefficient  $\rho$  and the allpass filter order. Features such as audio signal duplexing and automatic noise floor detection for sweep length determination will become available soon. The use of Ron Jeremy as an endorsing party is both gratuitous and unwarranted. As well, use of his name and likeness probably infringes on a number of copyright laws and was implemented with malicious disregard for basic moral decency.

## 1. INTRODUCTION

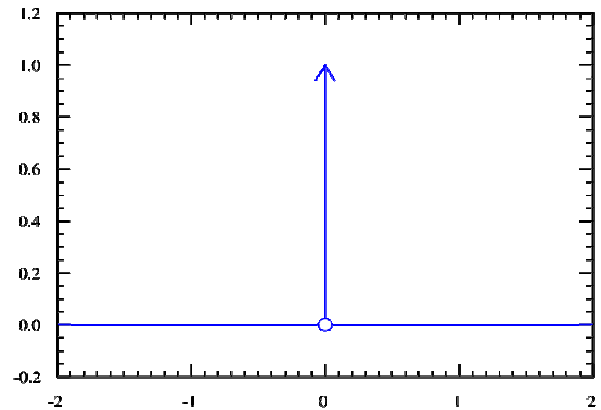
The purpose of this experiment was to generate a user friendly way of generating and processing acoustical characterization signals. These signals can be used to characterize audio processing equipment, reverberant spaces, acoustical insulation, microphones and loudspeakers.

Traditional acoustical measurements are carried out by playing a test signal containing all frequencies and recording the response. Therefore, there are three variables in a traditional acoustical measurement setup: the sound generation device (usually a speaker), the sound capture device (usually a microphone) and the space the sound travels through in between (almost always a room). By holding any two of these variables constant, the third can be experimentally measured.

### 1.1. Dirac-Delta Impulses

There are a variety of acceptable test signals that can be used in acoustical profiling. Each has specific advantages and disadvantages. The most traditional test signal is a Dirac-Delta impulse [1] (fig. 1, eq. 1).

$$\delta(x) = \begin{cases} +\infty, & x = 0 \\ 0, & x \neq 0 \end{cases} \quad (1)$$

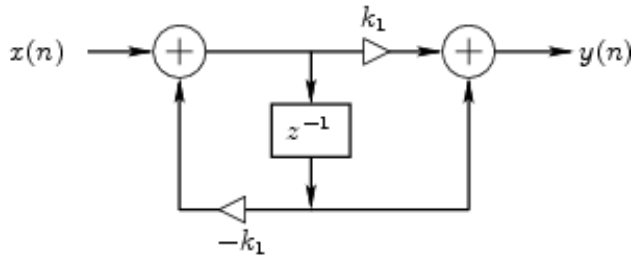


**Figure 1.** Dirac-Delta Impulse

Ideally, this signal has infinite amplitude distributed over an infinitely short period of time. Practically, this signal is impossible to generate. As well, speakers have a difficult time achieving high a SPL with such a short signal. This causes poor signal to noise ratio and can be damaging to the speaker.

### 1.2. All-Pass Deconvolution

Because the system is linear, the frequency content of an impulse can be spread out over time. This allows more energy to be put into room and increases SNR. By running an impulse through a series of all-pass filters [2](fig. 2), the frequency components of the impulse are spread over time according to the phase delay of the system [3].



**Figure 2.** First order all-pass filter in Direct Form II

The all-pass filter is calculated according to equation 2

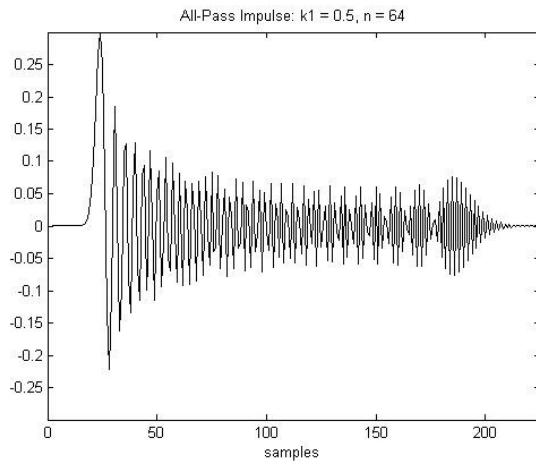
$$\left( \frac{k_1 + Z^{-1}}{1 + k_1 Z^{-1}} \right)^n \quad (2)$$

where  $n$  is the filter order and  $k_1$  is the all-pass filter coefficient. This filter can be implemented in Matlab according to equation 3.

```
GnA = [1 k1];
GnB = [k1 1];
h= [1, zeros(1, 1024)];
Gn = filter(GnB, GnA, h);
for i = 2:n;
    Gn = filter(GnB, GnA, Gn);
end;
```

(3)

An example all-passed impulse is given below in figure 4.



**Figure 3.** All-pass impulse  $k_1 = 0.5$ ,  $n = 64$

### 1.3. Swept Sine Measurements

While the smeared all-pass chirp introduces more energy into the system than a standard Dirac impulse, the signal energy is not maximized. More energy can be introduced by normalizing the signal at all time.

While this could be achieved with a compressor or some other signal processing techniques, it is easier to create normalized frequency swept sine waves using a fundamentally different set of equations. Both logarithmic (eq. 4, fig. 4) and linear (eq. 5, fig. 5) sine sweeps are commonly used to generate impulse response measurements [4]. Equations are given in Matlab code.

```
f=exp(log(20*(2/fs))*(1-[0:abins-1]'/abins));
phi = cumsum(pi*f);
X = sin(phi);
```

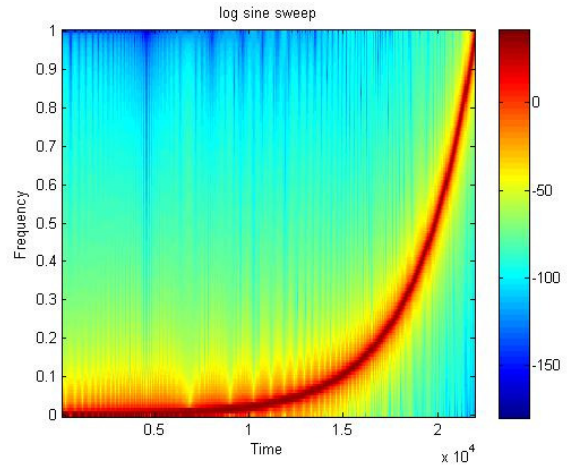
(4)

```
f = [0:abins-1]'/abins;
phi = cumsum(pi*f);
X = sin(phi);
```

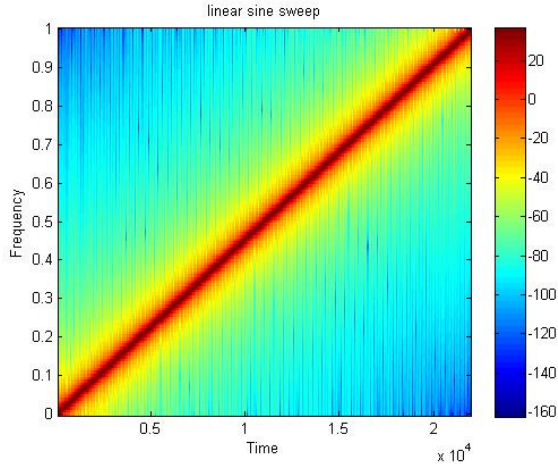
(5)

Where  $fs$  is the sampling rate and  $abins$  is the number of samples or the duration of the sweep.

Sine sweeps increase SNR by around 3dB per doubled sweep length. This allows for signals to be generated that accommodate for specific experimental conditions.



**Figure 4.** Logarithmic sine sweep spectrogram



**Figure 5.** Linear sine sweep spectrogram

#### 1.4. Golay Codes

A fourth commonly used acoustical measurement signal is the Golay codes [5]. These are signals composed of impulses that convolve to give a unit impulse with amplitude  $2L$ , where  $L$  is the order of the code. Each of the two Golay code sequences generated are  $2^L$  samples long. These signals were generated according to the Matlab code given in equation 6 [6].

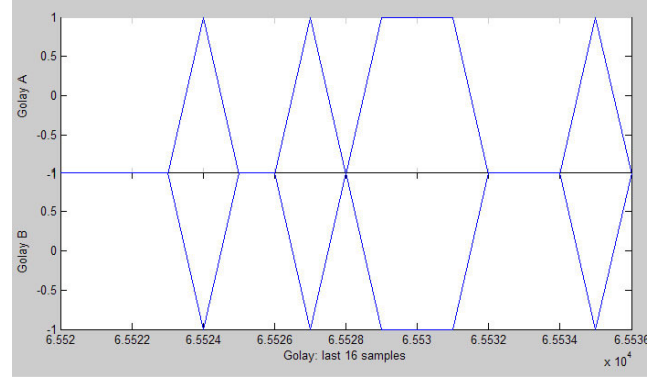
```

a = [1 1];
b = [1 -1];
while (N>1)
    olda = a;
    oldb = b;
    a = [olda oldb];
    b = [olda -oldb];
    N = N - 1;
End;

```

(6)

An example of the last 16 samples of order 16 Golay signals is given below in figure 6.



**Figure 6.** Last 16 samples of order 16 Golay codes

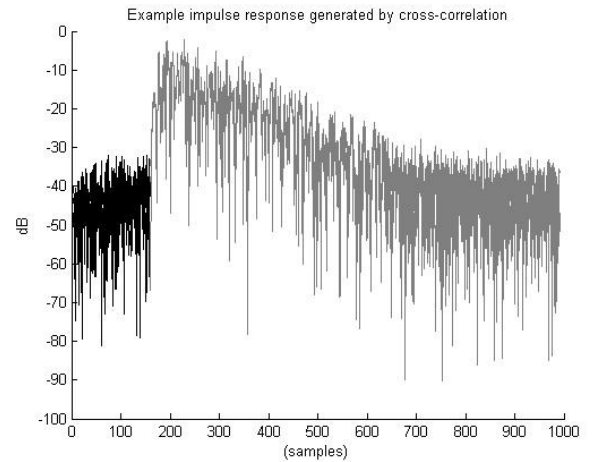
Golay codes give an excellent SNR. However, Golay codes have the distinct disadvantage of requiring that the two separate signals be played at different times. This imposes a strict requirement of time invariance on the test conditions.

#### 1.5. Signal Processing

Once any of these measurement signals are played and the system response is recorded, the processing routines are very similar across different test signals. The basic idea is that the original signal is cross-correlated with the measured response signal (eq. 7).

$$IFFT\left(\frac{FFT(response)}{FFT(input)}\right) = input \star response \quad (7)$$

This correlation or spectral deconvolution equates to division in the frequency domain [4]. This allows signals like the swept sine and all-pass chirp to “collapse” their frequency variance in the time domain into a traditional Dirac-like impulse response (fig. 7).



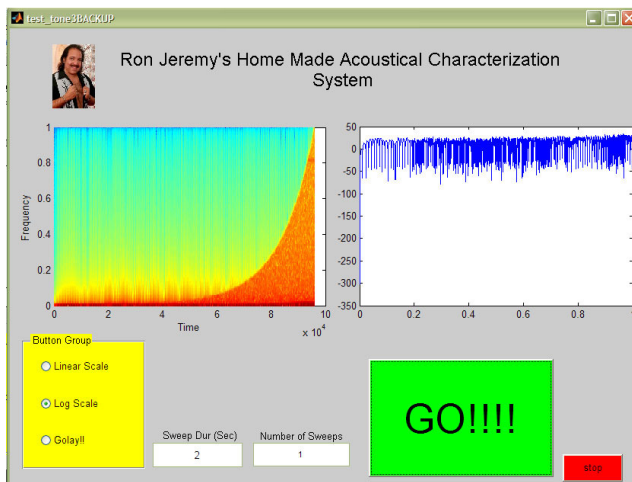
**Figure 7.** Example impulse response in dB

This processing routine can be accomplished using the Matlab commands *fftfilt* or *xcorr*.

When correlating responses to Golay codes, a slightly different processing routine is required. Golay codes require that the signals be convolved. This implies that the 'B' signal be inverted about the x-axis before correlation.

## 2. EXPERIMENTAL

Each of the previously mentioned signal generators were programmed in Matlab m-files. A GUIDE interface was then created (fig. 8) and the routines were able to be appropriately selected and called through a button group. Options such as Golay order, sweep duration, sample rate and sweep repetition were given on the interface as text input fields. The signals were then played through the soundcard while the response was simultaneously recorded. Duplexing was achieved through the use of MEX files (Windows only) given in the pa-wavplay bundle [7]. Graphical displays of the decorrelated response were given as both a spectrogram and a time domain signal.



**Figure 8.** Current interface design

## 3. RESULTS

So far, the program has achieved basic functionality. Although some of the processing routines are not giving the expected results, anticipated outcomes are expected after a few more weeks of tweaking and motherly nurturing.

## 4. DISCUSSION

In addition to fixing the aforementioned bugs in the program, an “auto-adjustment” feature could easily be included to automatically set the length of the swept sine wave signals. If a standard signal to noise ratio is set, the program could take a short sample of background noise and calculate a length assuming 3 dB of SNR improvement every time the sweep length is doubled. This should work well as long as the length doesn’t get too long.

If it is necessary, a profile of the speaker and microphone could be loaded into the program to allow for automatic compensation of the experimental measurement equipment.

For testing stereo microphone arrays or room tuning for surround sound setups, a standard polar plot of frequency specific response against absolute angle could be included. This would require the user to input an angle for each measurement. The polar response would then be updated at each successive measurement to give a 360° representation of the system response at a given frequency.

It may also be useful to include an option to display the graphs in a standard “figure” window to allow the user to perform basic graphical transformation functions such as zooming and tagging.

Ideally, once everything is working correctly, it would be nice to translate these functions to DX, AU or VST plugins to allow for signal generation and acquisition within a standard DAW. I imagine that this would make many of the standard operations much simpler because the DAWs are more specifically targeted to audio applications than Matlab.

## 5. REFERENCES

- [1] Wikipedia contributors, 'Dirac delta function', *Wikipedia, The Free Encyclopedia*, 5 May 2009, [http://en.wikipedia.org/w/index.php?title=Dirac\\_delta\\_function&oldid=288017888](http://en.wikipedia.org/w/index.php?title=Dirac_delta_function&oldid=288017888)
- [2] Smith, J. O. “Nested Allpass Filters,” *Physical Audio Signal Processing*, December 2008, [http://www.dsprelated.com/dspbooks/pasp/Nested\\_Allpass\\_Filters.html](http://www.dsprelated.com/dspbooks/pasp/Nested_Allpass_Filters.html)

- [3] Griesinger, D. "Impulse Response Measurements Using All-Pass Deconvolution," 1992. Retrieved from: <http://www.davidgriesinger.com/dgaes92b.pdf>
- [4] Farina, A. "Simultaneous Measurement of Impulse Response and Distortion with a Swept-Sine Technique," *Audio Engineering Society Convention, Preprint 5093*, 2000
- [5] Foster, S. "Impulse response measurement using Golay codes," *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '86.*, 1986, vol.11, pp. 929-932
- [6] Abel, J. and Berners, D. (2005). *Signal Processing Techniques for Digital Audio Effects*, 2005. Retrieved from: <http://ccrma.stanford.edu/courses/424/>
- [7] Frear, M. pa-wavplay, *SourceForge*, Sept. 2004. <http://sourceforge.net/projects/pa-wavplay/>