

BUILDING AN ACOUSTICAL-ENVIRONMENT MONITORING SYSTEM

**Jianxing Niu
Robert Barr
Wang Liao**

**Project Sponsor
Dr. Murray Hodgson**

**Engineering Physics 459
Engineering Physics Project Laboratory
The University of British Columbia
April 2nd, 2012**

**Project Number: 52
Group Number: 1216**

Executive Summary

The purpose of this project is to design and develop an acoustical-environment monitoring system. The prototype of this system is able to monitor ambient noise level, determine the impulse response and reverberation time of a room, calculate the speech transmission index (STI), and act as a sound masking system. The method used to achieve these goals is to continually monitor the amplitude of incoming sound, while also radiating low-level maximum length sequence (MLS) sound signals. The MLS signal received by microphones is then cross-correlated with the output signal in order to determine the impulse response of the room. The STI is determined with knowledge of the ambient noise level, impulse response, and a user defined speech level. To justify continuously emitting pseudorandom noise, the system also provides a sound masking function by filtering the emitted MLS to match the characteristics of masking sound. In doing so, the device meets the demand for privacy in buildings that lack a Heating, Ventilation, and Air Conditioning (HVAC) system such as newly developed sustainable buildings.

The prototype consists of a laptop, running MATLAB script, as well as the necessary peripherals such as microphones and loud speakers. A series of programs were written to monitor sound level, emit and filter MLS signals, measure the room response, calculate the STI, and determine the reverberation time for eight different octave bands. An A-weighting filter was also designed to adjust the measured ambient noise level in order to correspond to human hearing.

Our group has begun to lay the foundation for the project; however, there are still improvements to be made. The prototype's sound level measurement has been verified by a commercial sound level meter; however, due to time constraints, there has been a limited amount of testing for the impulse response, STI, and reverberation time measurements. It is recommended that future groups work to improve the timing of the impulse response function, perhaps by using a language or data-acquisition method that would allow for time-stamping the input and output signals, and run further testing. It is also recommended that future groups further test the quality of the sound masking system and rigorously determine the best MLS filter required for the task. It had also been found that MATLAB's built in function "xcorr" ran faster than the originally planned hadamard transform method [2] of cross-correlation. It is recommended that future groups look into this part of the code specifically. The hadamard transform method may have been necessary in the past, as [2] had been published in 1983 and assumes 1 microsecond per calculation, but there are possible alternatives including "xcorr" and "conv" functions in MATLAB. It is suspected that further optimization of the code would result from examining this issue. Of course, any further optimization by future groups would be beneficial to the project as well.

Table of Contents

Executive Summary.....	i
Figures.....	iii
Tables.....	iii
1. Introduction.....	1
1.1 Background and Significance.....	1
1.2 Project Objective.....	1
1.3 Scope and Limitations.....	1
1.4 Organization.....	2
2. Discussion.....	3
2.1 Theory.....	3
2.1.1 Maximum Length Sequence.	3
2.1.2 Sound Pressure and Sound Level Meter.....	4
2.1.3 Sound Masking System.....	5
2.1.4 Impulse Response.....	6
2.1.5 Speech Transmission Index.....	7
2.1.6 Reverberation Time.....	8
2.2 Methods.....	9
2.2.1 Impulse Response.....	10
2.2.2 Implementation.....	11
2.3 Experimental Equipment/Algorithms.....	14
2.4 Results.....	16
2.4.1 Sound Level Measurement.....	16.
2.4.2 Impulse Response.....	17
2.4.3 STI.....	18
2.4.4 Reverberation Time.....	18
2.5 Discussion of Results.....	19
3. Conclusion.....	20
4. Project Deliverables.....	21
4.1 List of Deliverables.....	21
4.2 Financial Summary.....	21
5. Recommendations.....	22
6. Appendices.....	23
Appendix A: MATLAB Scripts.....	23
Appendix B: Sound Masking System.....	28
Appendix C: Hadamard Transform.....	31
7. References.....	32

List of Figures

Fig.1. A simple circuit to generate maximum-length sequences.....	4
Fig. 2 A Sound Level Meter and its circular schematic.....	5
Fig. 3 An ideal Masking Spectrum of Sound Pressure level VS Octave Band Center Frequencies.....	6
Fig. 4 A Sample Impulse Response Measuring System	10
Fig.5 Relationship between input and output signal. $x(k)$ is the MLS noise input, $y(k)$ is the output, $h(k)$ is the impulse response	11
Fig. 6 Octave filters used for a 44.1kHz sampling rate.....	12
Fig. 7. A-weighting filter used for sound level meter.	13
Fig. 8. Experiment Set-up.....	14
Fig 9. A flow chart for the system.....	15
Fig 10. The final program's GUI.....	16
Fig 11. Sample of data taken during sound level function test. Octave bands range from 32 to 8000Hz.....	16
Fig 12. Impulse response of a large room.....	17
Fig 13. An example of an impulse response for a 32Hz octave band.....	17
Fig. 14 a). $RT_{60}=2.3696$ in noise space, on the bus.....	18
Fig. 14 b). $RT_{60}=0.4384s$ inside sponsor's office.....	18
Fig. 14 c). $RT_{60}=0.4384$ another measurement in sponsor's office.....	19
Fig. 15 A comparison of without and with sound masking system	28
Fig.16 Wiring Diagram of a two-channel sound-masking system with zone level..... controls, background music, and an amplified monitor panel.....	30
Fig. 17. A fast Hadamard Transform	31

List of Tables

Table 1. STI weighting factors proposed by Houtgast[15].	8
Table 2. Financial Summary.....	21

1. Introduction

This section outlines the significance of the project. It provides the motivation for development at the time of writing, the overall objective, and the scope and limitations the group has faced.

1.1 Background and Motivation

Green buildings, which are environmentally friendly and resource-efficient throughout their life-cycle, are gaining popularity across the world. A key principle of green buildings is the protection of occupant health and the increase of productivity. As a result, monitoring systems are widely used to detect the light, heat, and air-condition in the building. The acoustic properties of these buildings, however, are rarely monitored. Furthermore, because many green buildings are naturally ventilated, there is an absence of ambient noise caused by Heating, Ventilation, and Air Conditioning (HVAC) systems, which leads to a reduction in speech privacy. Dr. Murray Hodgson from UBC has proposed to make an attempt to address these concerns.

1.2 Project Objective

The objective of this project is to lay the foundation for developing a system that has the capability to do the following:

- Monitor the ambient noise level
- Adjust the measured noise levels to correspond to human hearing (A-weighting)
- Compute the impulse response of a room
- Calculate the speech transmission index (STI)
- Calculate the reverberation time
- Generate MLS signals with the spectrum of sound masking signals

1.3 Scope and Limitations

The original scope, as detailed in project objectives, is mostly achieved. An MLS signal of any order can be generated and filtered according to sound masking spectrum before emission. While the room's noise level can be sampled at a user-defined rate; and an A-weighting filter then adjusts the measured loudness to reflect sensitivity of human hearing. The impulse responses are then obtained and split to eight different octave bands. Based on the impulse responses, the STI and reverberation times can be calculated.

The project has, however, been limited due to the cons of using MATLAB. While the

strengths of MATLAB, specifically the signal processing functions, have greatly reduced production time the environment has limitations in multitasking. For example, when computing the impulse response of a room the program must both emit MLS signals and record audio from a microphone. This can be achieved by creating "audiorecorder" and "audioplayer" objects; however, it is difficult to find an accurate relationship between the beginning of playing and the beginning of recording. This may lead to a misleading portion of the impulse response curve prior to the point corresponding to the time the initial signal reaches the microphone. There may be a solution to this in the data acquisition (DAQ) toolbox or through dedicated DAQ hardware however.

The proposed STI calculation was changed and expanded because of the existence of interfering noise. When the user enters a desired speech level, the program will calculate the signal to noise ratios for eight different octave bands, the corresponding modulation functions, the apparent signal to noise ratios for different octave bands, the different modulation frequencies, and finally the STI for eight separate octave bands and weighted STI result.

Further limitations can be attributed to a lack of time. Little progress has been made to determine the effectiveness of the sound masking system. The original plan included consideration for speaker and microphone placement as well as a study of different MLS filters; however, time restrictions did not permit any quantitative analysis.

1.4 Organization

This report will first present some theories regarding MLS, impulse response, STI calculation and reverberation time. Then, the report will discuss in detail about methods, set-up, findings and results. Conclusions and recommendations will also be included.

2. Discussion

This section contains detailed theory relevant to this project, the methods used to implement and develop the system, and the set-up and flowchart for the system. Primary test results will be presented and discussed.

2.1 Theory

2.1.1 Maximum Length Sequence (MLS)

1) Definition

A maximum length sequence is a cyclical chain of numbers which has certain well-defined properties. They are periodic and can reproduce every binary sequence. They are certain classes of linear recurring sequence and can also be referred to as m-sequences

2) Generation:

- It can be generated by using a basic shift register and a modulo-two gate. Certain feedback loop should also be used during the design process.
- It can also be generated by software if we need a reproducible signal.

3) Properties [10]

- The amplitude of the noise is in the band of frequencies; it is not a fixed parameter. This is simply obtained by choosing the correct combination of clock pulse period and the number of shift register stages so that the “flat” part of the power spectrum lies in the required frequency range
- The amplitude and total power of sequence may be controlled precisely by choosing the parameters of the sequences
- These properties are not affected by environmental factors such as temperature or time.

4) Application

MLS signals are crucial in the control engineering field, where the impulse response of a system may be found by introducing low-level signals into it. This impulse response may be determined by cross-correlating the signal input and the system output, and making the use of the fact that the cross-correlation of the input is known to be a constant.

5) Specific circuit design

Maximum length sequence can be generated either by hardware or by software. The hardware approach is included here. In this project, MLS is generated by MATLAB

programs.

One of the simplest circuits for generating maximum-length sequences is as below

- The nth and kth ($K < N$) stages are fed to the modulo gate, which is an adder
- On the application of a shift pulse the digits in the shift register move one stage to the right, and the sum digit enters the first stage
- The output, which may be taken from any of the stages (although we normally think of the output as coming from the last stage of the register), is a pseudo-random digit sequence

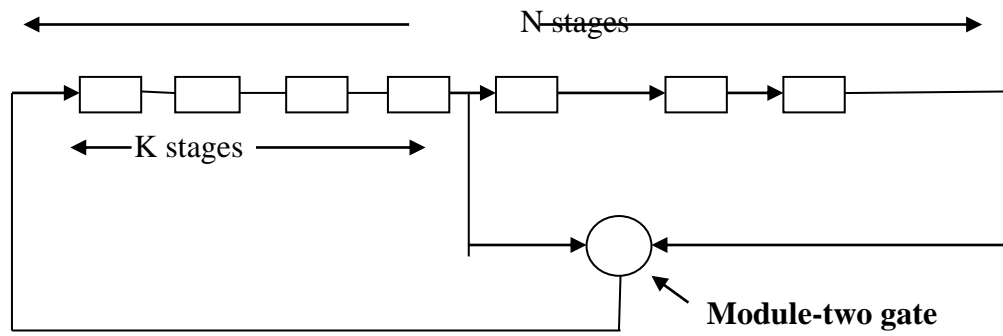


Figure 1. A simple circuit to generate maximum-length sequences [10]

2.1.2 Sound Pressure and Sound Level Meter

Sound pressure is the local pressure deviation from normal atmospheric pressure caused by a sound wave. The sound pressure level (L_p) is a logarithmic measure of effective pressure [11]. The formula for computing sound pressure level (L_p) is given by

$$L_p = 10 \log_{10} \left(\frac{p_{\text{rms}}^2}{p_{\text{ref}}^2} \right) = 20 \log_{10} \left(\frac{p_{\text{rms}}}{p_{\text{ref}}} \right) \text{ dB},$$

where P_{rms} is the rms sound pressure being measured; P_{ref} is called the reference sound pressure, equal to $2 \times 10^{-5} \text{ (N/m}^2\text{)}$, which is usually considered as the threshold of human hearing.

In industry, Sound Level Meters are widely used to measure the amplitude of ambient noise. A sample circular schematic is shown in Figure 1, which is composed of an amplifier, comparator and LED for different sound level. Sound level meters usually have different weighting-filters to adjust the measured sound level. Even though the Sound Level Meter has been proven as a working device, it is most suitable for use in industry area and it usually only gives an instantaneous measurement making it difficult to determine a person's average exposure to noise

over long time.

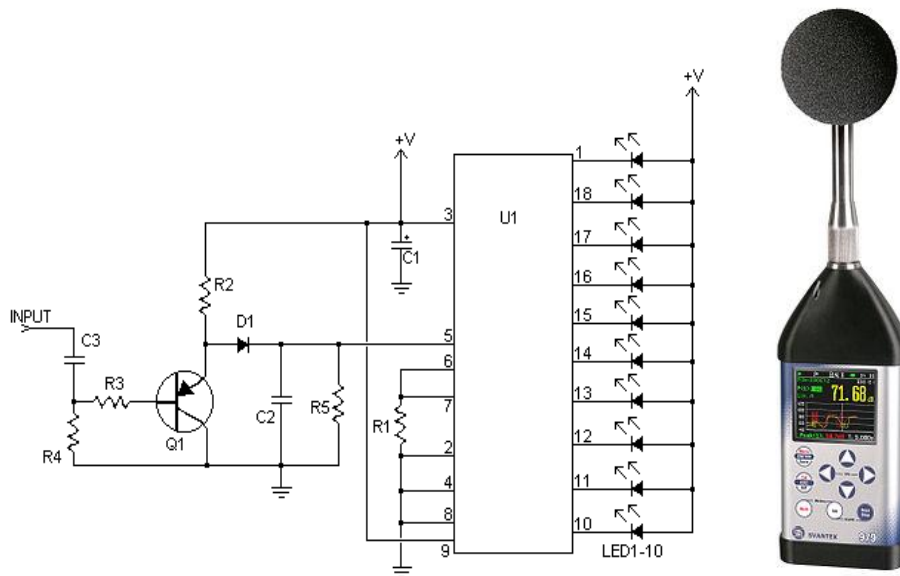


Fig. 2. A Sound Level Meter and its circular schematic [12]

2.1.3 Sound Masking System

1) Introduction

A sound masking system emits low-level, non-distracting masking noise designed to reduce speech intelligibility and thereby improve speech privacy. It can be of great value in open-plan offices, doctor's examination rooms and other environments where confidentiality is important. The goal is to increase the sound level of a masking spectrum above the building or office noise level.

Some commercial sound masking systems use special low-level masking noise to improve speech privacy and confidentiality. The noise is usually non-obtrusive.

2) Masking Spectrum

The masking sound level should be above the background noise level but cannot be too high to be obtrusive. It's found that the masking sound level for the listener should be between 45 and 50 dB [13]. One way to achieve this level for a walled space or ideal open-plan space (5-foot minimum partition, absorption ceilings, etc) is to use the masking spectrum plotted below [13]. The solid line is the ideal sound pressure level over frequency and the dotted lines show the tolerance.

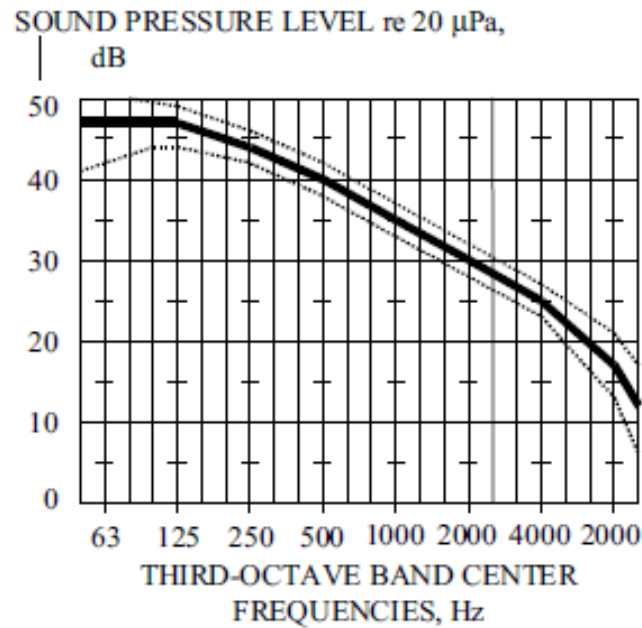


Fig. 3 An ideal Masking Spectrum [13] of Sound Pressure level VS Octave Band Center Frequencies.

2.1.4 Impulse Response

An important measurement in order to determine the acoustical characterization of a room is the impulse response. The system will be placed in a room of UBC CIRS building, which can be viewed as a linear and Time Invariant system characterized by an impulse response $h(t)$. Because many acoustical parameters such as reverberation time can be derived from impulse response, it's important to measure binaural room impulse responses with a very high signal-to-noise ratio.

A common method for measuring the impulse response of such an acoustical system is to apply a known input signal and to measure the system's output [13]. The excitation signal and the deconvolution technique enabling the obtention of the impulse response from the measured output should satisfy the following requirements:

- The emitted signal must be perfectly reproducible
- The excitation signal and the deconvolution technique have to maximize the signal-to-noise ratio of the deconvolved impulse response

For this project, our sponsor, Dr. Murray Hodgson, recommends the use of Maximum-Length Sequences (MLS) as the input signal. With the maximum length sequence method, it is possible to measure the impulse response with a great amount of accuracy and repeatability. One advantage of the MLS method is its cross-correlation technique, which is a measure of similarity of two waveforms as a function of a time-lag applied to one of them. The MLS method is not sensitive to many kinds of stationary noise as several measurements can be averaged. This

property makes the method desirable for acoustical measurements in noisy environments [15].

Another advantage to the MLS method is that the signal could be filtered to have the spectrum of sound masking noise. This would result in a non-obtrusive technique for continuous measurement.

2.1.5 Speech Transmission Index

Speech Transmission Index (STI) is a numeric representation of speech transmission quality in a communication channel. The STI value varies from 0 (poor) to 1 (excellent). STI is influenced by speech level, talker to listener distance, background noise level, and impulse response.

Because the S/N may be a function of frequency, the octave-band specific approach of calculating STI is used. Farrel Becker [16] and Houtgast [17] described a detailed procedure to calculate speech transmission index from the impulse response:

- 1) Filter the overall impulse response to seven octave bands with center frequencies at 125Hz, 250Hz, 500Hz, 1000Hz, 2000Hz, 4000Hz, 8000Hz. Square the impulse response for each octave band
- 2) Integrate the squared impulse responses to get the total energy
- 3) Perform Fourier transform on the squared impulse response. This gives the envelope spectrum
- 4) Take the absolute value of the Fourier Transform results.
- 5) Normalize the envelope spectrum by dividing the FTT of impulse response by the total energy from 2).
- 6) Calculate the Modular Transfer Functions (MTF) as in Eqn (1). For each of the octave band, MTFs take the amplitude at 14 modulation frequencies F spaced 1/3 of an octave apart from 0.63Hz to 12.5Hz.

$$m(F) = \frac{|\int_0^\infty e^{-2\pi i F t} * r(t) dt|}{\int_0^\infty r(t) dt} * \left(1 + 10^{\frac{-S}{10}}\right)^{-1} \quad (1)$$

In the case of continuous interfering noise, $m(F)$ is reduced by a reduction factor $\left(1 + 10^{\frac{-S}{10}}\right)$, where S/N is the signal to noise ratio.

- 7) Convert each of the 98 modulation values to 98 apparent signal-to-noise ratios (S/N).

$$\left(\frac{S}{N}\right) = 10 \log\left(\frac{m}{1-m}\right) \quad (2)$$

- 8) Limit the S/N range to 30dB. Any value greater than 15dB is set to 15dB and any value less than -15dB is set to -15dB
- 9) Calculate the mean apparent S/N for each octave band

10) Compute 7 octave band STI values using equation (3)

$$STI_{oct} = \frac{\left(\frac{S}{N}\right) + 15}{30} \quad (3)$$

11) Finally, the overall STI is the weighted mean of octave-band specific STI. The weighting factors used here are proposed by Houtgast[17], as tabulated below.

Table 1. STI weighting factors proposed by Houtgast[17]

Center Octave-band Freq (Hz)	Weighting Factor
125	0.13
250	0.14
500	0.11
1000	0.12
2000	0.19
4000	0.17
8000	0.14

2.1.6 Reverberation Time

The reverberation time is an important parameter for characterizing the quality of an auditory space. In specific, the reverberation time of the room specifies the duration for which a sound persists after it has been switched off. It is normally defined as T_{60} , which is the time taken for the sound to decay to 60dB its value at cessation. One important reason of knowing the reverberation time is its closer relationship with the speech transmission index (STI). The shorter the reverberation time, the larger the STI will be, which indicates the good listening quality of the room in which the measurement has been performed.

Our main method for measuring the reverberation time is to record a speech like test signal and play it back. In this case, we can use a MLS signal as our test signal.

The theoretical value of the reverberation time depends on the geometry dimension of the room and the absorptive characteristics. The specific equation is given below:

$$\text{Reverberation time } RT_{60} = k \cdot V / A$$

Where the factor $k = (24 \cdot \ln 10) / c_{20} = 0.049$ (feet) and with $k = (24 \cdot \ln 10) / c_{20} = 0.161$ (meter).

V = room volume in m^3

$A = \alpha \cdot S$ = equivalent absorption surface or area in m^2

α = Absorption coefficient or attenuation coefficient

RT_{60} = reverberation time in s
 S = absorbing surface in m^2
 $A = \alpha_1 \cdot S_1 + \alpha_2 \cdot S_2 + \alpha_3 \cdot S_3 + \dots$
 c_{20} = Speed of sound is 343 m/s or 1126 ft/s at 20 °C

Even though this equation has been proved to be very accurate, sometimes we may not have the dimension or material information of the office, chamber or hall room that we want to perform the measurement.

Therefore, the method we used here was to derivate the reverberation time by integrating the maximum length sequence impulse response. It also known as Schroeder's method. Specifically speaking, RT is determined from decay curves obtained by radiating sound into the test enclosure. The decay curve is the received signal following the cessation of the sound source.

Let $n(t)$ be a stationary white noise source with power σ^2 per unit frequency, and $r(t)$ be the impulse response of the system consisting of the receiver, transmitter, and the enclosure. Then a single realization of the decay curve $s(t)$ from the interrupted noise experiment is given by

$$s(t) = \int_{-\infty}^0 n(\tau) r(t - \tau) d\tau,$$

where the noise is assumed to be switched off at $t=0$, and the lower limit is meant to signify that sufficient time elapsed for the sound level to reach a steady state in the enclosure before it was switched off. The reverberation time is obtained from the decay curve $s(t)$

2.2 Methods

Figure 5 shows a cheap, adaptable and measurement system which is composed of a computer with a MATLAB program that generates an MLS signal [14]. The MLS signal is used as source signal ($x(k)$), which is a pseudorandom binary sequence consisting of +1 and -1 (the 1 generated by the shift-register is mapped to -1, and 0 is mapped to 1). A specific circuit consisting of shift registers and an adder for generating MLS sequence are in the Appendix.

The MATLAB program generates the MLS sequence, and the MLS sequence is radiated through the loudspeaker of the sound masking system connected to the power amplifier. The response will be measured by the microphone and recorded through the soundcard. A few measurements will be averaged to increase accuracy and the data will be analyzed by a computer program to give impulse response, reverberation time, and noise level. The time required for one measurement is very short, only a few

seconds.

In order to accurately measure the impulse response of a room. The sampling frequency, number of averages (i.e. number of times and emitted signal will be sent to the loudspeaker) and recording mode (mono or stereo) is decided by the user.

For the MLS sequence, the order (also the number of stages used in the shift register) m will be made variable, making the number of samples contained in one period of an m order MLS sequence (ie. The periodic length of MLS) $L=2^m-1$, and the sampling frequency will also need to be set. Many measurements can be averaged in order to improve SNR as a MLS is deterministic and will average to itself, while background noise will average to zero.

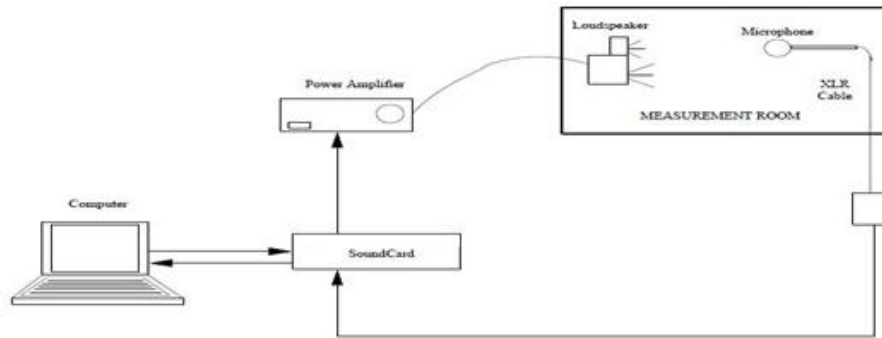


Fig. 4 A Sample Impulse Response Measuring System [14]

2.2.1 Impulse Response

Figure 6 shows a sample input, output and impulse response [15]. There is a cross-correlation between the input $x(k)$ and the output $y(k)$ of a linear system. It can be shown that this is related to the auto-correlation of the input by a convolution with the impulse response:

$$R_{xy}(k) = R_{xx}(k) * h(k)$$

The MLS has an advantageous property that its auto-correlation function is essentially an impulse, represented by the *Dirac delta function*:

$$R_{xx}(k) \approx \delta(k)$$

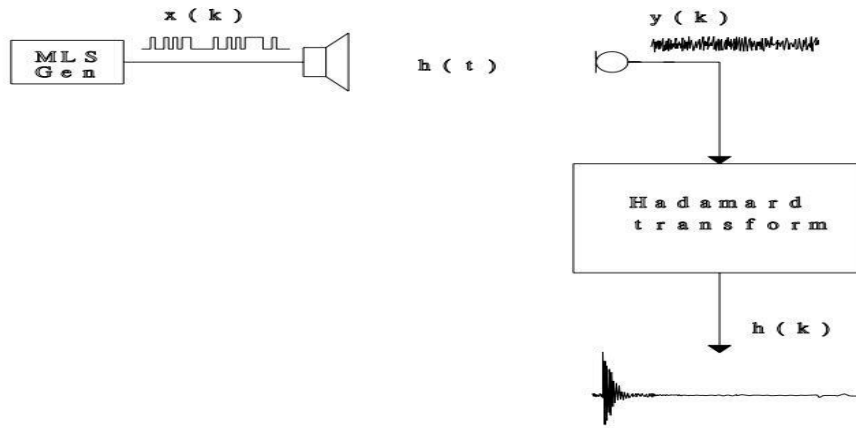


Fig.5 Relationship between input and output signal. $x(k)$ is the MLS noise input, $y(k)$ is the output, $h(k)$ is the impulse response [15].

The result of convolving a sequence with a *Dirac delta function* is the sequence itself. Thus the impulse response $h(k)$ can be found by cross-correlating the noise input $x(k)$ with the output $y(k)$:

$$R_{xy}(k) \approx \delta(k) * h(k) = h(k)$$

Because MLS is a pseudorandom noise signal, the simplest way to calculate this cross-correlation is using the fast Hadamard transform, which can be done by using the MATLAB function Hadamard, and it only requires $2.5n\log_2(n)$ operations of additions and subtractions.

The effect of Hadamard transformation is to transfer the individual responses at different times and in different directions so they appear as a response to impulse appearing simultaneous and in one direction.

Although MLS has similar spectral properties to true random white noise, it is actually a deterministic signal. This determinism gives the advantage that the signal can be repeated precisely. It is therefore possible to increase the signal-to-noise ratio (S/N) by synchronous averaging of the response sequences $y(k)$. Any extraneous, uncorrelated background noise will then be reduced for each averaging as long as the noise is "stationary".

Although the Hadamard transform method may be the most efficient method, upon implementation it was found that MATLAB's "xcorr" function ran more efficiently. Due to time constraints, this could not fully be explored and will be left to future groups for testing.

2.2.2 Implementation

2.2.2.1 MLS

The MLS is used as an excitation signal and is emitted from a loudspeaker. To mask sound, a filter is designed to adjust the amplitude of the generated MLS in different octave bands based on the masking spectrum. As a result, the MLS signal is louder at low frequency octave band and quieter at high frequency octave band.

A microphone samples the ambient noise. When the measurement is finished, the response $y(k)$ is cross-correlated with the original signal $x(k)$ to find the impulse response $h(k)$ as in figure 6. This impulse response is then divided into octave bands and can be used for STI and reverberation time measurements.

2.2.2.2 Sound Level Meter

The incoming signal is first multiplied by a calibrated constant and then filtered into octaves. The range of center frequencies for the octaves depend on sampling rate. For example, a sampling rate of 44.1 kHz will result in a range of 32 to 8000Hz octave bands as the next highest, 16kHz, contains frequencies over the Nyquist rate of 22.05kHz as seen in fig 7.

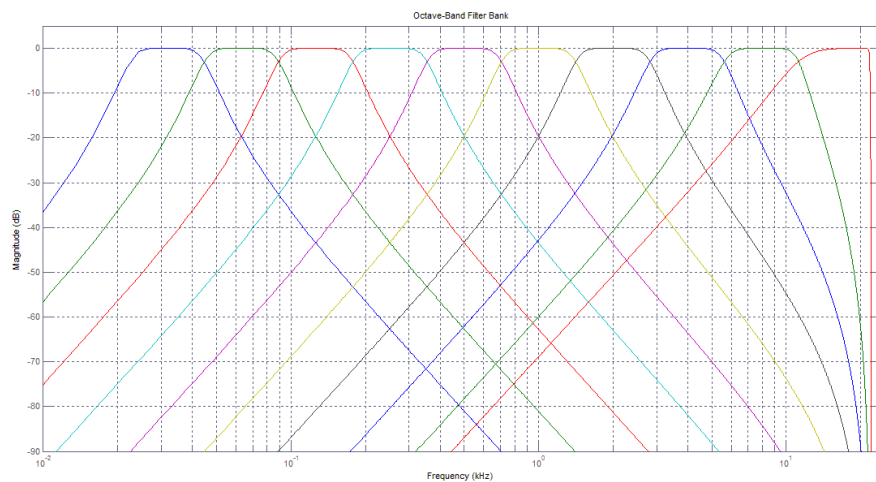


Fig. 6 Octave filters used for a 44.1kHz sampling rate

Once in octave bands, the signal is then squared, summed, and divided by its length. These values are then converted into decibels, i.e. $10 \cdot \log_{10}(\text{values}/\text{Pref}^2)$. This results in the decibel value for each octave band.

If A-weighting is desired, the signal will first be passed through an A-weighting filter as seen in fig 8 before being passed to the octave filters. The final result is the decibel value of each octave band corresponding to human hearing.

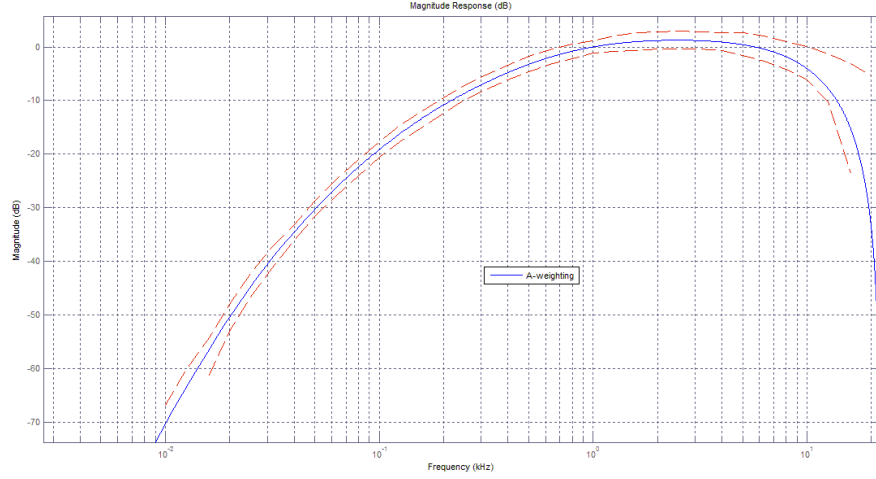


Fig. 7. A-weighting filter used for sound level meter.

2.2.2.3 STI Calculation

STI is calculated as described in section 2.1.

2.2.2.4 Reverberation Time Calculation

Schroeder [9] has shown that the expected response $r(t)$ to a continuous noise excitation switched off at time $t=0$ is related to the impulse response $h(t)$ by the following equation:

$$E[r^2(t)] = G \int_t^\infty h^2(u) du$$

where G is a constant related to the excitation level. On the other words, the reverberation decay curve can be found by differentiating the impulse response distribution of test signal. The impulse response is more like the sound intensity decay form. In our program, we just approximately fit a straight line on the impulse response to indicate the decay of sound intensity, and find out the reverberation time by finding the slope of the fit straight line.

2.3 Experimental Equipment/Algorithms

An external Roland cakewalk (UA-25EX) sound card and a RION sound level meter (NA-29E) are used to facilitate calibration. The output from the sound level meter microphone is sent to both the meter and the sound card. The values calculated by both the program and the sound level meter can then be compared.



Fig. 8. Experiment Set-up

A flowchart for the system is shown in figure 9. The system was initially developed as separate programs and later combined with a GUI. For easier understanding, the functions here are listed as separate entities to avoid confusion with handles passed between user input and the programs. The GUI can be seen in figure 10. For further comments see appendix A.

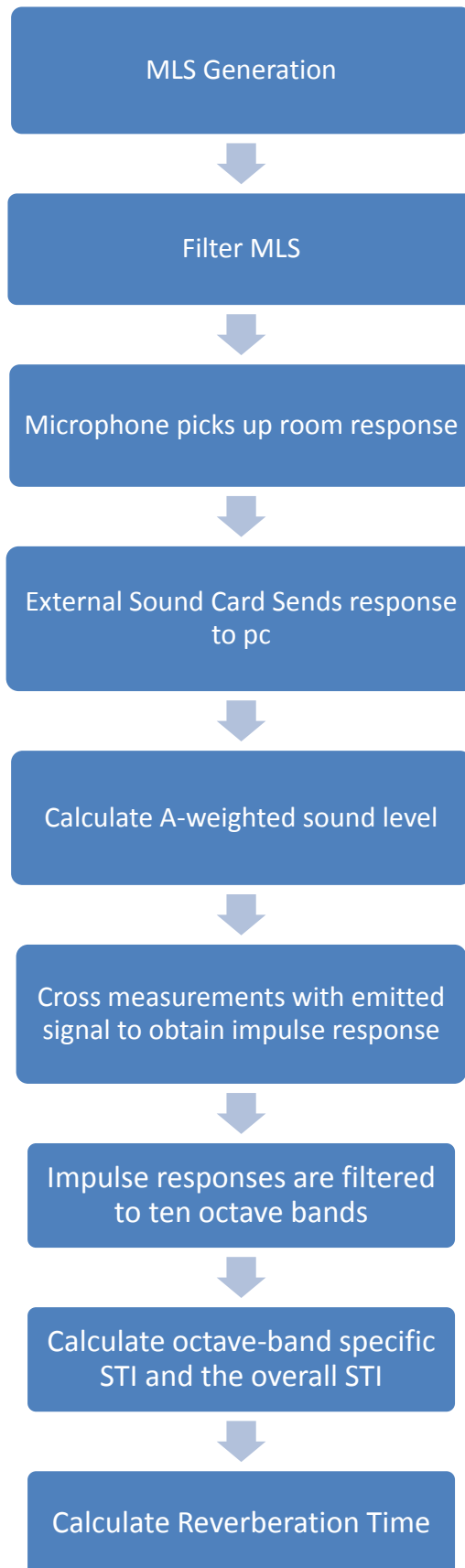


Fig 9: A flow chart for the system

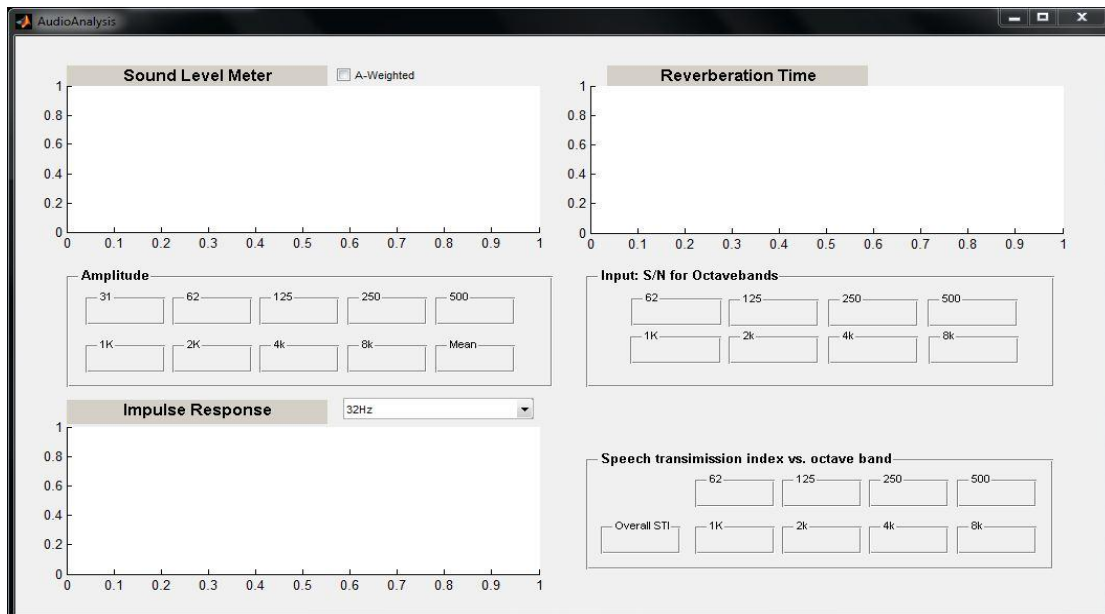


Fig 10: The final program's GUI

2.4 Results

2.4.1 Sound Level Measurement

Measurement data was collected and sent to a commercial sound level meter (Rion) and our program. After the incoming data was multiplied by a linear calibration constant, the program gave very accurate results. In figure 11 a sample of the recorded values are given.

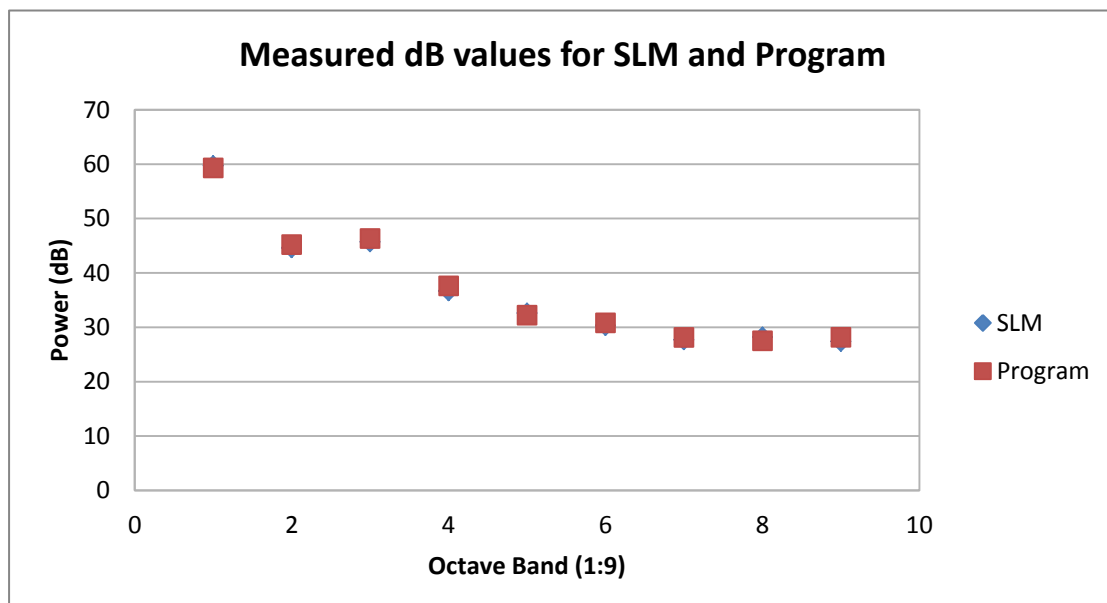


Fig. 11. Sample of data taken during sound level function test. Octave bands range from 32 to 8000Hz

The greatest difference in values observed was around 0.95 decibels. The sampling

times were not perfectly synchronized, however, so any difference could be due to a slight change in actual sound level. Therefore, this difference is negligible. The A-weighting curve was also tested in this way with similar results.

2.4.2 Impulse Response

Several samples were taken to measure impulse response. Regrettably, due to time limitations we were not able to set up the commercial product to compare data however. An example of the data taken can be seen in figure 12. This sample had been taken in a large room so it is expected that there would be a quick decay. It is also important to note that the sample has a longer delay in front of the initial impulse than desired as the timing in the function is not perfect. The sample covers all octaves.

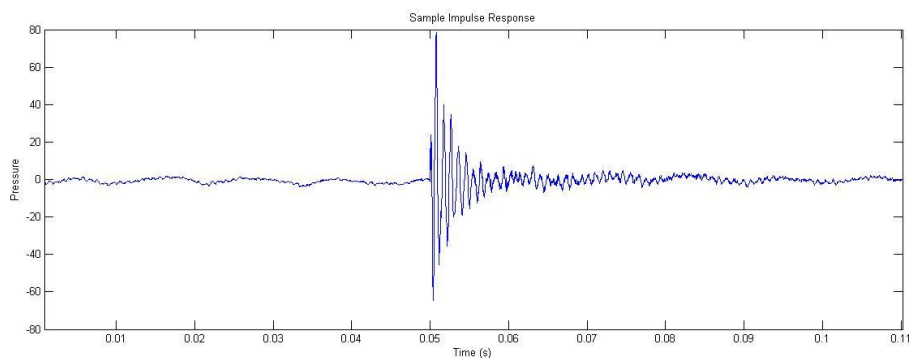


Fig. 12. Impulse response of a large room

The function also divides measured impulse responses into octave bands. An example of this performed in a smaller room can be seen in figure 13. This figure contains the impulse response of a 32Hz octave band. Note that time has been shifted to remove delay in the signal.

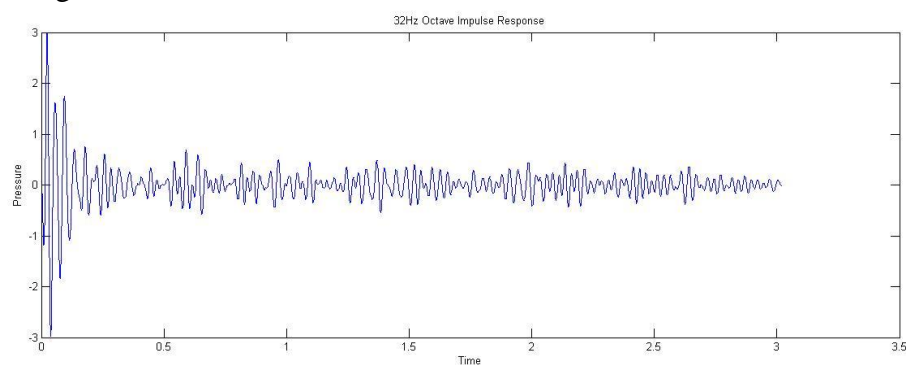


Fig. 13. An example of an impulse response for a 32Hz octave band

It should be noted that these measurements have not been calibrated or tested alongside a commercial product.

2.4.3 STI

The STI is calculated based on seven octave-band specific impulse responses. A test run shows the overall STI to be 0.6 in the sponsor's lab. Although not compared with commercial software yet, the STI is not accurate because the room's speech intelligibility appears to be good.

2.4.4 Reverberation Time

Below are two kinds of main reverberation time measurement we made from both the sponsor's office, the UBC Sustainability building and on the bus.

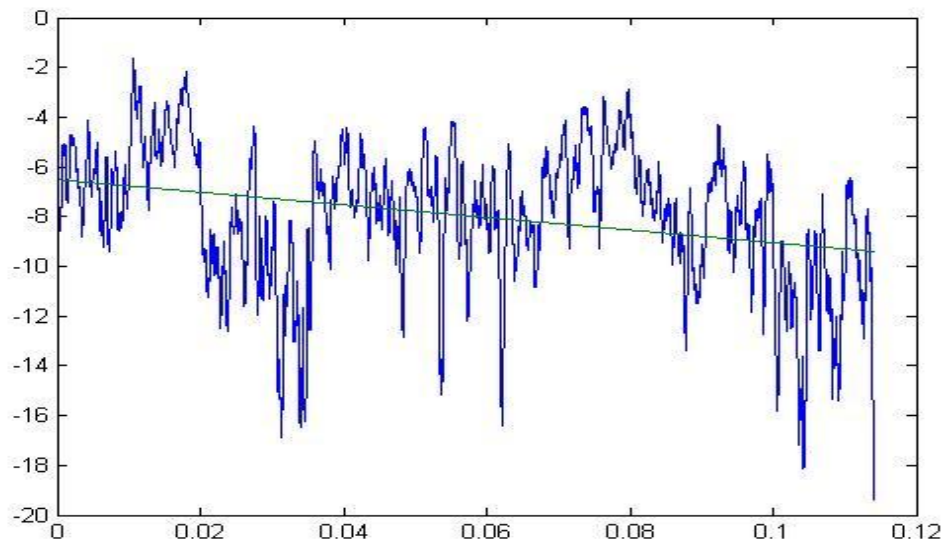


Fig. 14 a). $RT_{60}=2.3696$ in noise space, on the bus

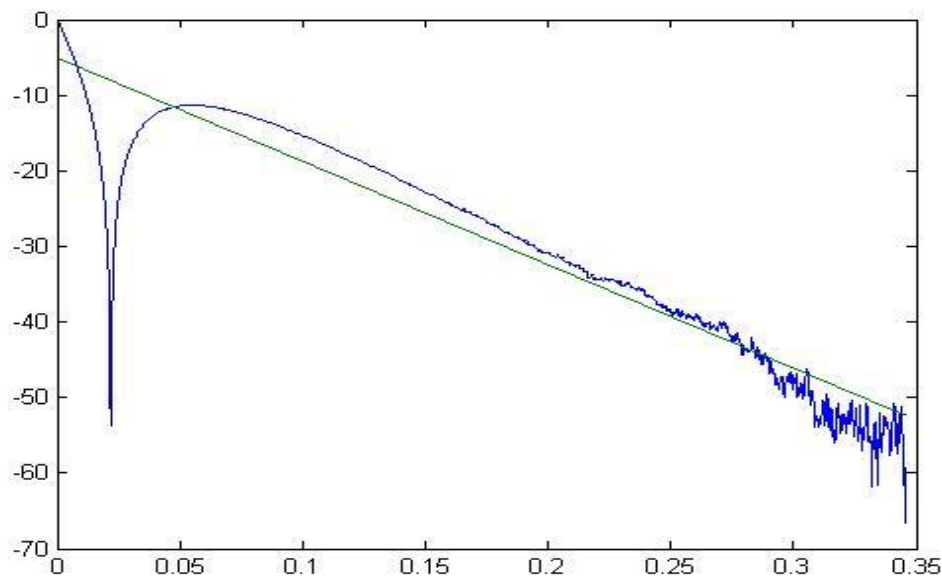


Fig. 14b). $RT_{60}=0.4384s$ inside sponsor's office

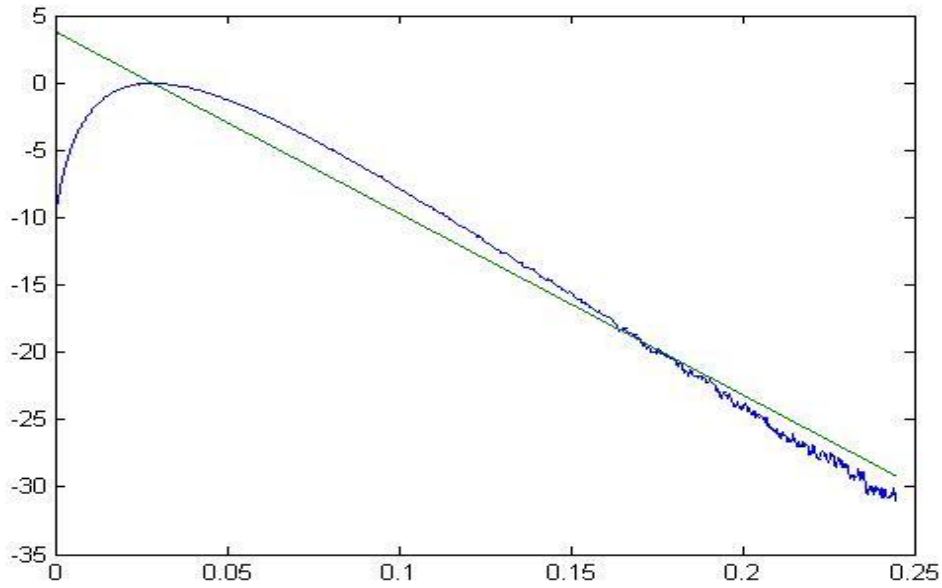


Fig. 14c). $RT_{60}=0.4384$ another measurement in sponsor's office

The time that the sound intensity has begun to decay is counted as soon as the noise generator is turned off. From the graphs above, we can find out that the reverberation time in UBC sustainability building are around half seconds, which is low enough to provide a good Speech Transmission index level.

The test result is proved to be similar to the result provided by our sponsor, which is around 0.5s.

2.5 Discussion of Results

Regrettably, due to time constraints, we were not able to rigorously test this system. Much more testing is needed before these results can be trusted. Unfortunately, that task must be left to a future group.

3. Conclusion

The necessary components for the acoustical-monitoring system have been developed. The system is able to generate a maximum length sequence of any order, and filter this signal to correspond to a sound masking's spectrum. The signal can be emitted continuously through a loudspeaker. Meanwhile, a microphone keeps measuring the room response and an external sound card samples signal at a user defined rate. A program then cross-correlates the output with the input by use of the "xcorr" function. The obtained impulse responses are then filtered to octave bands. Seven octave-band specific impulse responses are used to calculate octave band STIs, which are weighted and summed to achieve the overall STI. The reverberation time is also calculated.

The sound level measurement program has been calibrated and found to be in agreement with the commercial product. Due to time constraints, the STI and reverberation time have not been verified yet. Still, this project has made a step towards proving the concept of using filtered MLS signal to measure acoustic parameters while also masking sound to improve speech privacy.

4. Project Deliverables

4.1 List of Deliverables

- An Engineering Recommendation Report
- A working prototype
- Documented results from testing
- Instruction/documentation for the system

Most proposed deliverables are accomplished except test results. It was proposed to test the system at different rooms of different acoustical conditions. This has not been achieved.

4.2 Financial Summary

Table 2. Financial Summary

Name	Cost	Funded by
LabVIEW 2011 Student Edition	\$65	Project Lab

5. Recommendations

This section lists and details possible recommendations for future groups. The recommendation focuses on testing and improvement.

- **Verify Impulse Response, STI, and Reverberation time.** Test the system at multiple locations with different acoustic conditions. Compare the measurements with a commercial product
- **Optimize the MATLAB scripts.** Due to the large amount of sampled data, the STI calculation takes few minutes. This could be improved with the optimization of code.
Also, it is suspected that the cross-correlation function could be improved. The planned hadamard transform function did not match the efficiency of MATLAB's "xcorr" function. Further research on this topic is needed. Another possibility is the "cconv" function.
- **Improve the timing for playing and receiving MLS signals.** This could possibly be achieved using DAQ hardware or the MATLAB DAQ toolbox.
- **Improve the sound masking effect.** Though the MLS has been filtered based on sound masking spectrum, it still sounds obtrusive and distracting. It's strongly recommended for future groups to ensure the signal to be less disruptive.
- **Test speaker and microphone placements.** To optimize sound masking effect, the speaker needs to be placed at a proper location so the masking signal will drop by 5dB at the listener's location.
- **Find out the minimum hardware requirement.** Based on primary test results, the processing power requirement appears to be high. It's suggested to find out the minimum hardware requirement. There is a trade-off between the system performance and hardware.
- **Develop an embedded system.** A dedicated microcontroller with peripherals to emit, receive and analyze signals will be more portable, power-efficient and cost-effective.

Appendix A: MATLAB Scripts

The following sections are descriptions of code before being combined into a single program with GUI. Ask the project sponsor for fully commented code.

1) **AudioAnalysis**

This is the complete project and GUI it displays values as seen in fig 10

2) **AmbientMain**

This was the original function that accessed and looped the functions below for testing. All persistent variables can be modified at the top of the function.

3) **recordAudioData(recordTime, Fs, nBits, nChannels, ID, C)**

This function records the initial audio sample before the main loop begins. It creates and returns an audio sample and a recording object.

recordTime = the length in seconds of the recording

Fs = Sampling frequency (8k as 8000, 11.025k, 22.05k, 44.1k, 48k, 96k, etc.)

nBits = Sample bit depth (8,16,24)

nChannels = 1 for mono, 2 for stereo

ID = recording device ID (-1 for default, use audiodevinfo(IO, ID) to find others)

C = Constant to map voltage to pressure for calibration

4) **DesignOctaveFilters(Fs)**

This function designs and returns a vector of df2sos values for use with the filt() function. Each set corresponds to an octave filter of different center frequency. For example, entering an Fs value of 44100 will yield a vector of values corresponding to 32, 64, 125, 250, 500, 1k, 2k, 4k, 8k Hz octave bands.

Fs= sampling frequency

5) **filteroctaves(OctaveBands, myRecording)**

This function filters myRecording with octave band filters designed in DesignOctaveFilters and finds the dB value of each octave. It returns a vector of values corresponding to the power value of each octave filtered.

OctaveBands = vector of df2sos values

myRecording = sample to be filtered

6) **DesignAfilter(Fs)**

This function designs an class "A" A-weighting filter. It returns a dfilt.df2sos value for use in the filt() function.

Fs= Sampling Frequency

7) AWeighting(Afilter, myRecording)

This function returns an A-weighted audio sample.

Afilter = A weighting df2sos value

myRecording = audio sample to be filtered.

8) ambientNoiseAnalysis(myRecording)

This function returns the Fourier transform the recording in dB and Power as well as the number of unique points in the sample below the Nyquist rate as well as the number of points used in the fast Fourier transform. It is unused in the final project.

9) plotAmbientAnalysis(myRecording, Fs, dbSpectrum, NumUniquePts, nfft, PowerSpectrum, octpower)

This function was mainly used for testing. It makes several subplots of the ambient analysis.

10) testMLS

This was the main function for testing the impulse response measurements. It plays and records audio and performs all correlation, octave filtering and plotting.

Variables as defined at top of code.

11) genMLS(polynomial)

Takes in the polynomial representation of an MLS and creates the sequence. Returns the sequence and length of the sequence.

polynomial = [a b c] where a is the number of registers (ie length= 2^a-1), and b and c where the registers are tapped. (Or polynomial = $x^a+x^b+x^c$)

12) designMLSfilt(Fs)

This designs the filter for the sound masking system. It returns a df2sos value.

Fs=sampling frequency

13) MLSfilt(pnSequence,MLSfilter)

This function filters the MLS for the sound masking system.

pnSequence = MLS

MLSfilter = df2sos value

14) InvMLSfilt(myRecording, InvMLSfilter)

This function performs the inverse filter to the received audio sample.

InvMLSfilter= df2sos value

15). STI(impulse response, sn_ratio)

This function inputs the seven octave-band specific impulse responses and the S/N ratio, then calculate 98 modulation transfer functions, which then give 98 apparent S/N ratios and hence 7 octave-band specific STI. The STIs are then weighted to give a final overall STI.

16). Reverberation Time

This matlab code calculates the reverberation time of our test signal, MLS sequence in this case. The two input variable are the (record signal 's' and the sampling frequency 'Fs')

The final output is display on line 69.

Comment about reverberation time code:

Although the reverberation time test result is about the same as the test result provided by the sponsor, the code still needs further improvement.

- a) Sometimes the code cannot detect the reverberation time from the test signal, it may be the reason of the test position differences.
- b) The octave band filter needs to be installed to find out the reverberation time at different octave band. For now, it can only measure the reverberation time over one period.

Code for reverberation time:

```
15) function r = rt60 (s, Fs)
16) % rt60 (s, Fs) : determine reverberation time from signal s with sampling
17) % frequency Fs (44100 Hz for most wav's).
18)    Fs=44100;    % Sampling frequency (8k,11.025k,22.05k,44.1k,48k,etc)
19) nBits=16;      % Sample bit depth (8,16,24)
20) nChannels=1;    % Number of Channels 1=mono,2=stereo
21) ID=-1;         % Input device ID; -1 for default; Use audiodevinfo
22) recordTime=10; % Length of recording
23) C=10000;       % Constant to map sample data to noise data
24) s=recordAudioData (recordTime,Fs,nBits,nChannels,ID,C);
25) %%    initialize the octave band
26) BandsPerOctave = 1;
27) N = 6;         % Filter Order
28) F0 = 1000;     % Center Frequency (Hz)
29) % Fs = 44100;   % Sampling Frequency (Hz)
30) f = fdesign.octave(BandsPerOctave,'Class 1','N,F0',N,F0,Fs);
31)
32) F0 = validfrequencies (f);
```

```

33) Nfc = length (F0);
34) for i=1:Nfc,
35)     f.F0 = F0 (i);
36)     OctaveBands (i) = design (f,'butter');
37) end
38) t = (0:(length (s)-1))/Fs;
39) dt = 1/Fs;
40)
41) s2 = abs (s);
42) s2 = s.^2;
43) s2db = 10*log10 (s2/max(s2)); %% approximate decibal
44)
45) startThreshold = -10;
46) endThreshold = median (s2db);
47) for n=length (OctaveBands):-1:1
48)     yw=filter (OctaveBands (n),s);
49)     s2db (:,n)=yw
50) end
51) yw = ma (s2db, floor (Fs/2200)); %% convolution
52) tart = 1;
53) t1 = 1;
54) t2 = 4;
55) r = [];
56) while ~isempty(t1) && ~isempty(t2)
57)     t1 = start + find (s2db (start:end) > startThreshold,1, 'first');
58)     t2 = t1 + find (s2db (t1:end) < endThreshold, 1, 'first');
59)     start = t2;
60)     if (t2-t1)*dt < 0.1
61)         continue
62)     elseif ~isempty(t1) && ~isempty(t2)
63)         myenv = s2db (t1:t2);
64)         tt = ((1:length (myenv))*dt)';
65)         H = [tt ones (length (myenv), 1)];
66)         myenv (isinf (myenv)) = 0; % todo fix this
67)         thta = H\myenv(:);
68)
69)         r = [r, -60 / thta(1)];
70)         plot (tt, myenv, tt, thta (1)*tt + thta (2))
71)         % pause
72)     end
73) end
74) r (n)=[length (r) mean (r) std (r) median (r)]
75) disp (sprintf ('Analyzed % d sounds, mean RT60 = % f +/- %f, median RT60 = % f, length (r),
mean (r), std (r), median (r)))

```

76)

77)

78) function r = ma (s, N)

79) r = conv (s, ones (N, 1)/N); %% convolution impulse response

80) r = r (1:length (s));

Appendix B: Sound Masking System

There are three steps to design a masking system, as listed below

a) **Attenuate the Direct Sound**

Ensure the sound is not transmitted from the talker to the undesired listener directly.

b) **Reduce Sound Reflections**

Minimize the reflections of sound when transmitted from the talker to the undesired listener.

c) **Raise the Ambient Sound Level Using Sound Masking**

Sound masking improves the speech privacy by adding low-level background noise, hence reducing the speech-to noise ratio and speech intelligibility.

Well-designed room acoustics is an important component of a successful masking system. As described by Taylor, Hoover and Keith [13], partitions and screens can be used to attenuate the direct sound; absorptive furniture and ceilings can reduce sound reflections; a low level of random electronic noise is also used to mask the remaining sounds, reducing its intelligibility and increasing speech privacy. Some commercial sound masking systems are available. Figure 2 shows a comparison of an office with no sound masking system and an office with an installed sound masking system that used all three steps [13].

In the scope of this project, MLS is emitted as low-level background noise in order to raise the ambient noise level, mask sounds and also measure acoustic parameters.

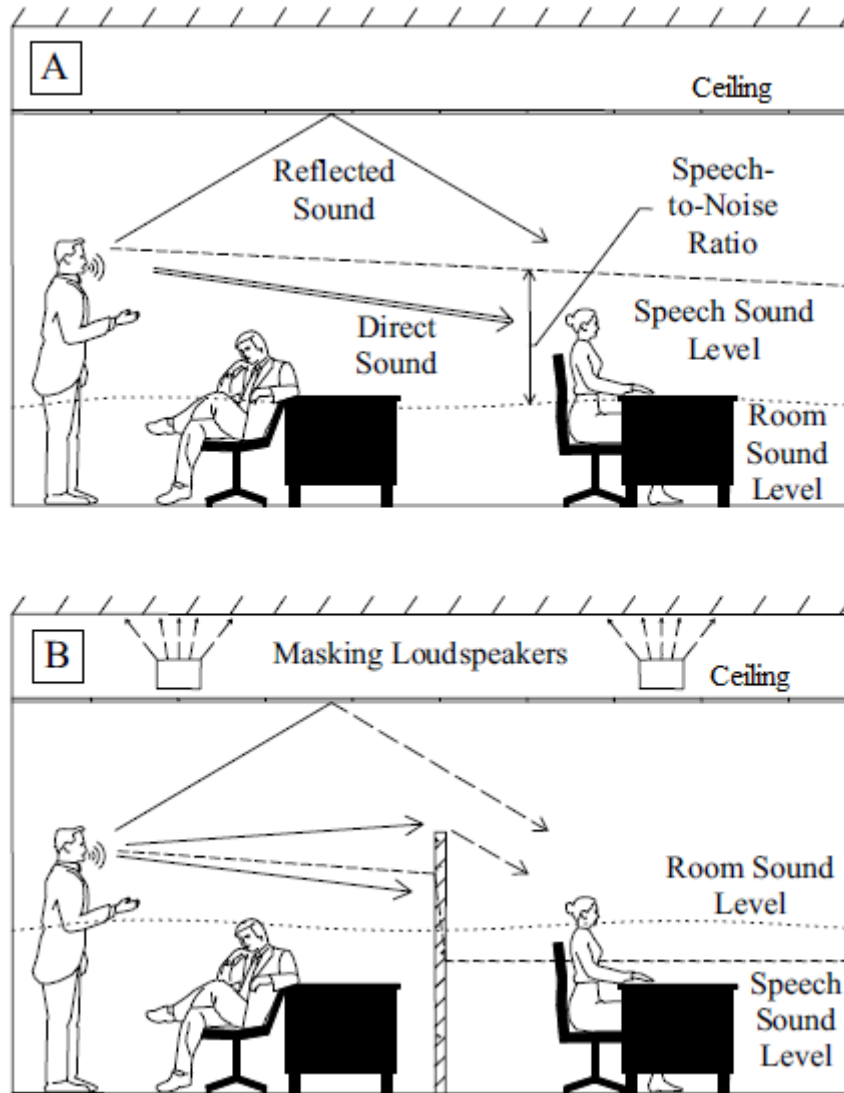


Fig. 15. A comparison of without and with sound masking system [13]

2). Electrical Components of the Masking System

A typical sound masking system consists of a masking noise generator, an equalizer, one or more power amplifiers and a group of special loudspeakers installed above a dropped ceiling [13]. A masking system may be one-channel, two-channel, or multi-channel. A one-channel sound-masking system is the simplest, however, it has the limitation that all loudspeakers give coherent sound, hence when a person walks from one zone of loudspeaker to another, he will hear the phase cancellation and a reduced masking effect. A two-channel or multi-channel masking system solves this problem by using separate noise generator and equalizer for different zones. A wiring diagram for a two-channel sound-masking system is shown as Fig. 3 [13].

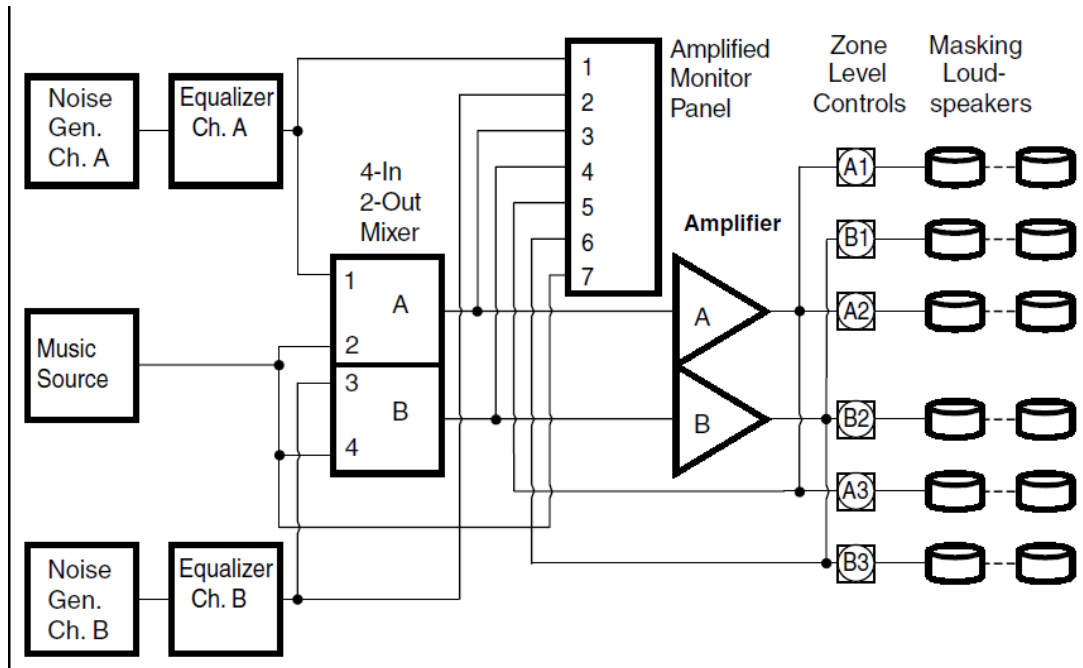


Fig.16 Wiring Diagram of a two-channel sound-masking system with zone level controls, background music, and an amplified monitor panel

Appendix C: Hadamard Transform

1). Definition

It is a $2^m \times 2^m$ matrix and can be regarded as being built out of size-2 discrete Fourier transforms, and is in fact equivalent to a multidimensional DFT of size $2 \times 2 \times 2 \dots \times 2$.

2). Hadamard matrix Equation:

$$H_m = \frac{1}{\sqrt{2}} \begin{pmatrix} H_{m-1} & H_{m-1} \\ H_{m-1} & -H_{m-1} \end{pmatrix}$$

$$(H_m)_{k,n} = \frac{1}{2^{\frac{m}{2}}} (-1)^{\sum_j k_j n_j}$$

3). Fast Hadamard Transform Algorithm for maximum length sequence impulse response

- Write a matrix form $y = Mh$, with $y = (y_0, y_1, \dots, y_{L-1})$; $h = (h_0, \dots, h_{L-1})$; $M_{ik} = m_{i+k}$, where $L = 2^n - 1$, n is a big integer
- M is a left circular matrix of order L and all cyclic shifts of the sequences are found in the rows as well as in the columns of M
- $H = 1/(L+1) (M_y (y, \dots, y))$; $y = \sum_{j=0}^{L-1} y_j$

4). One major advantage of Hadamard Transform is that it requires fewer operation time, only $n \log_2(n)$ operations.

There are many different algorithms for calculating a “fast” Walsh-Hadamard Transform. The following is one of the most elegant:

- a- Construct $N+1$ column, each with 2^N rows.
- b- Place the input vector $x[n]$ in the 1st column.
- c- Moving left to the right, fill in the next column as follows:
In the top half of this column, place the pairwise, mutually exclusive sums of the previous column; in the bottom half, place the pairwise differences and repeat for step c of the remaining columns.

x	a	b	y
x[0]	$a[0]=x[0]+x[1]$	$b[0]=a[0]+a[1]$	$y[0]=b[0]+b[1]$
x[1]	$a[1]=x[2]+x[3]$	$b[1]=a[2]+a[3]$	$y[1]=b[2]+b[3]$
x[2]	$a[2]=x[4]+x[5]$	$b[2]=a[4]+a[5]$	$y[2]=b[4]+b[5]$
x[3]	$a[3]=x[6]+x[7]$	$b[3]=a[6]+a[7]$	$y[3]=b[6]+b[7]$
x[4]	$a[4]=x[0]-x[1]$	$b[4]=a[0]-a[1]$	$y[4]=b[0]-b[1]$
x[5]	$a[5]=x[2]-x[3]$	$b[5]=a[2]-a[3]$	$y[5]=b[2]-b[3]$
x[6]	$a[6]=x[4]-x[5]$	$b[6]=a[4]-a[5]$	$y[6]=b[4]-b[5]$
x[7]	$a[7]=x[6]-x[7]$	$b[7]=a[6]-a[7]$	$y[7]=b[6]-b[7]$

Fig. 17. A fast Hadamard Transform [18]

References

- [1] J. Borish, "Self-Contained Cross-correlation Program for Maximum-Length Sequences," *Journal of the Audio Engineering Society*, vol. 33, pp. 888-891, Nov, 1985.
- [2] J. Borish and J. B. Angell, "An Efficient Algorithm for Measuring the Impulse Response Using Pseudorandom Noise," *Journal of the Audio Engineering Society*, vol. 31, pp. 478-488, July/Aug, 1983.
- [3] W. T. Chu, "Impulse Response and Reverberation Decay Measurements Using a Periodic Pseudorandom Sequence," *Applied Acoustics*, vol. 29, pp. 193-205, 1990.
- [4] H. Date and K. Fukudome, "Observation of the Envelope of Reverberation Curves by Using Two Relatively Prime Pseudorandom Sequences," *Acustica*, vol. 62, pp. 291-299, 1987.
- [5] B.P. Flannery, W.H. Press, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipes in FORTRAN 77: The Art of Scientific Computing*, pp. 287-290. New York: Cambridge University Press, 1992.
- [6] M. Holters, T. Corbach and U. Zölzer, "Impulse response measurement techniques and their applicability in the real world," in *Proc. of the 12th Int. Conference on Digital Audio Effects (DAFx-09)*, Como, Italy, 2009, pp. 1-5.
- [7] E. Mommertz and S. Müller, "Measuring Impulse Responses with Digitally Pre-emphasized Pseudorandom Noise Derived from Maximum Length Sequences," *Applied Acoustics*, vol. 44, pp. 195-214, 1995.
- [8] D. Rife and J. Vanderkooy, "Transfer Function Measurements with Maximum-Length Sequences," *Journal of the Audio Engineering Society*, vol. 37, pp. 419-444, June, 1989.
- [9] M. R. Schroeder, "New Method of Measuring Reverberation Time", *J. Acoust. Soc. Am.* 37 (1) pp. 409 – 412 (1965)
- [10] W.D.T Davies, "Generation and properties of maximum –length sequence", *Imperial Chemical Industries Ltd*
- [11].Sound Pressure [Online]. Available FTP:
http://en.wikipedia.org/wiki/Sound_pressure#cite_note-1
- [12] Sound Level Meter [Online]. Available FTP:
<http://www.aaroncake.net/circuits/vumeter.asp>

- [13] Ashton Taylor, "Sound Masking System", Hoover & Keith Inc. Houston, TX.
- [14] G. Stan, J. Embrechts and D. Archambeau, "Comparison of Different Impulse Response Measurement Techniques," *Journal of the Audio Engineering Society*, vol. 50, pp. 249-262, Dec, 2002.
- [15] Ole Herman Bjor, "Maximum Length Sequence", Gracey & Associates, Cheleveston, UK.
- [16] Farrel Becker. A Do-it-Yourselfers Guide to Computing the Speech Transmission Index [Online]. Available FTP:
<http://www.synaudcon.com/site/articles/a-do-it-yourselfers-guide-to-computing-the-speech-transmission-index/>
- [17] T.Houtgast, H.J.M.Steeneken and R. Plomp, "Predicting Speech Intelligibility in Rooms from the Modulation Transfer Function. I. General Room Acoustics"
- [18] Rama Ratnam, Douglas L. Jones, Bruce C. Wheeler, William D. O'Brien, Jr., Charissa R. Lansing, and Albert S. Feng "Blind estimation of reverberation time" 18 August 2003
- [19] Art Ludwig. "Part III: Maximum Length Sequence" [Online]. Available FTP:
http://www.silcom.com/~aludwig/Signal_processing/Maximum_length_sequences.htm
- [20] M. Cohn & Lempel, "On Fast M-Sequence Transforms", IEEE Trans. Inf. Theory IT-23pp

Reference Summaries:

[1] This article elaborates on the author's previous paper [2] by suggesting the basis for a cross-correlation program that computes permutation vectors during the permutation operation as opposed to pre-computing and storing vectors as used in most algorithms. This approach sacrifices speed for lesser memory requirements; however, the difference in speed is very small because the approach also reduces the disk access time resulting from accessing pre-computed data. The approach will create a self-contained program that will ease the process of implementing with hardware.

[2] This article describes a method of using pseudorandom noise to measure the response of a linear system. This can be done very efficiently when a maximum length sequence is used as noise because the binary sequence eliminates multiplications from the problem. The fast Hadamard transform can be used to minimize additions and eliminate subtractions. When using this approach the number

of additions is approximately $(2.5n)\log_2(n)$ where n is the length of the sequence. This approach will lead to a superior dynamic range as opposed to using impulse excitations.

[3] This paper presents experimental data for using cross-correlation of a periodic pseudorandom sequence to measure impulse response and reverberation decay. The experimental data shows the methods potential for being used in noisy environments. The cross correlation method has been shown to produce a high signal-to-noise ratio. In addition there is the possibility of using long sequences and averaging results. The method has also been shown to be capable of measuring attenuation or transmission loss.

[4] This paper focuses on measuring reverberation time in low signal-to-noise environments. The method uses two pseudorandom sequences whose periods are relatively prime. A signal is formed by switching one quick driven modified pseudorandom sequence with the other slow driven sequence emitted from a speaker. The received envelope is cross-correlated with the switching sequence. This output is equal to the envelope of the squared impulse of the area between the speaker and the microphone. It is also shown that the signal-to-noise ratio is proportional to the period of the sequence. This paper focuses mainly on theory, then some computer simulation to confirm results.

[5] The section of this book used, "Generation of Random Bits", focuses on generating single random bits using low level programming languages or chips. Two methods are shown. One is useful for hardware applications, while the other is more useful for machine language.

[6] This paper outlines three techniques for measuring impulse response. The techniques use either maximum length sequences (MLS), exponentially swept sines (ESS), or time delay spectrometry (TDS). The paper aids in choosing the best technique for different circumstances. It is shown that the MLS technique can be performed with very efficient algorithms and therefore has little computational demands, but is susceptible to nonlinear distortion outside of linear amplitude range. The ESS method is not as sensitive to nonlinear distortion and allows for higher measurement amplitudes which reduces the time needed to reach a desired accuracy. Also, the accuracy of the ESS method is more suited to the range of human hearing. However, the ESS method cannot make measurements under a nonzero "lowest" frequency and the method has a high computational demand as a convolution must be calculated. The TDS method does not require correlation or Fourier transforms, however, it can only measure impulse signals. The authors offer a correction to this, however, afterwards there are no real advantages over the other methods.

[7] This paper goes over the method of using the Hadamard transform and maximum-length sequences to measure impulse response. It also mentions using a

pre-emphasis filter with band-pass dependant time separating properties as a band-pass filter.

[8] This paper gives a very comprehensive analysis of transfer function measurement based on maximum length sequences. It shows that the MLS approach can as effective as time delay spectrometry at rejecting nonlinear noise in weakly nonlinear environments. The paper defines a coherence function and demonstrates that the MLS technique can be cost effective.

[9] This paper compares four different methods of measuring impulse response. These methods are maximum length sequence (MLS), inverse repeated sequence (IRS), time-stretched pulses (TSP), and logarithmic sine sweeps. The paper shows that in noisy environments the MLS and IRS techniques are preferred and in quiet environments the logarithmic sine sweeps are preferred.