
Measurement Laboratory at Home - Exercise 2

Student Name(s): Taeyoung Kim

Term: SS2023

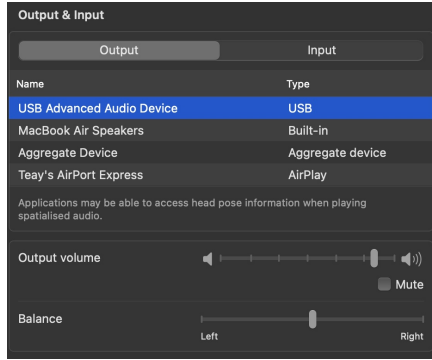
Date: May 21, 2023

The purpose of this report is to explore and familiarize ourselves with the USB Audio Interface (UAI) through various exercises. The UAI is a crucial component in audio systems, providing the means to connect audio devices to a computer for signal input and output. In this report, we will delve into generating and recording signals, calibrating the UAI's inputs and outputs, and analyzing latency and potential jitter.

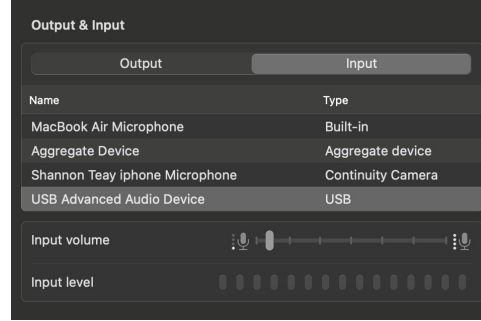
The exercises in this report aim to enhance our understanding of the UAI and its associated hardware and software-related factors that contribute to latency. Latency refers to the delay experienced between the time a signal enters a system and the time it emerges. This delay can be introduced by a range of elements, including A/D and D/A converters, filters, and buffers, among others.

1 Check that the UAI runs correctly

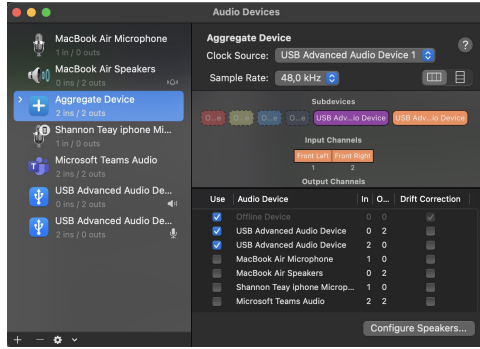
This section focuses on verifying the functionality of the USB Audio Interface (UAI). Through a series of checks, we ensure that the UAI is connected properly to the computer and that any required drivers or software are installed. We then proceed to test the audio playback and recording functionalities of the UAI to ensure that it is functioning correctly. This verification step is crucial in establishing a reliable foundation for the subsequent exercises and analyses conducted in the report. The UAI was connected to a MacBook laptop using a USB hub for the verification process. Figure 1 on the following page illustrates the initial settings of the Audio, USB Audio Interface (UAI), and Breadboard used in the verification process.



(a) Output Audio Setting



(b) Input Audio Setting



(c) Connect Aggregate Audio



(d) Connect UAI and Breadboard with Laptop using hub

Figure 1: Comprehensive display of the initial settings

2 The First Measurement

Throughout this section, white noise in the range of $[0, 0.5]$ was generated for a duration of 5 seconds using the **rand** function. The signal was sampled at a rate of 48 kHz (SampleRate), and buffer sizes of 1024 (BufferSize) were utilized. To play and record the signal simultaneously, the **play_rec(aPR, sig)** function was employed.

2.1 Assignment 1

In this MATLAB implementation, the **rand()** function is used to generate white noise in the range of $[0, 1]$, which is then scaled by 0.5 to meet the desired range $[0, 0.5]$. The audioplayer and audiorecorder objects are used to handle the playback and recording operations. The play function starts

playing the white noise, and the record function starts recording simultaneously. After waiting for the specified duration, the stop function is used to stop the recording. The recorded signal is then retrieved using the **getaudiodata** function, and it can be processed or plotted as desired.

2.1.1 Plotting recData and playData over time

On the following Figure 2 illustrates both the input (recData) and output (playData) signals for channel 1 and channel 2 over time with white noise.

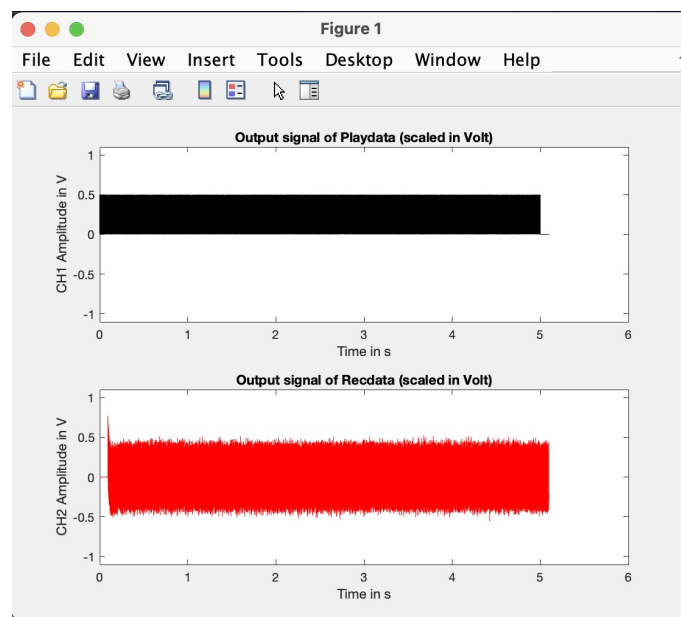


Figure 2: Plotting **recData** and **playData**

2.1.2 Understanding the Time Discrepancy between Played and Recorded Signals

This question refers to the observed time shift or delay between the played (output) signal and the recorded (input) signal. It explores the factors that contribute to the time difference, such as system latency, buffer delays, and the inherent delay in audio hardware and software components involved in simultaneous playback and recording.

The time shift observed between the played and recorded signals can be attributed to factors such as system latency and buffer delays. When audio is played and recorded simultaneously, there is inherent latency caused by the

audio hardware and software components involved. This latency refers to the delay between sending audio data for playback and receiving the recorded audio data. The latency can vary based on the specific hardware, drivers, and settings used.

In addition to system latency, buffer sizes used during playback and recording introduce further delays. Larger buffer sizes, such as the 1024-sample size mentioned in the code example, can contribute to additional delay as audio data is processed in chunks.

The cumulative effect of system latency and buffer delays leads to the noticeable time shift between the played and recorded signals. To minimize this shift, adjusting the buffer size or optimizing audio settings can be helpful. By estimating the system latency and compensating for it, the recorded signal can be aligned with the played signal.

2.1.3 Shifted Mean Levels: Investigating the Discrepancy between Played and Recorded Signals

When the played and recorded signals exhibit a shift with respect to their mean levels, it suggests the presence of a DC offset or bias in the audio hardware or software. This offset refers to a constant voltage or value added to the audio signal, resulting in a deviation in the average level of the signal.

DC offsets can arise from imperfections or calibration issues within audio systems, particularly in components like the audio interface or analog-to-digital converter (ADC) employed for recording.

The discrepancy observed between the played and recorded signals in terms of their mean levels implies a difference in the DC offset between the playback and recording paths. Consequently, it leads to a noticeable alteration in the baseline or average level of the signal.

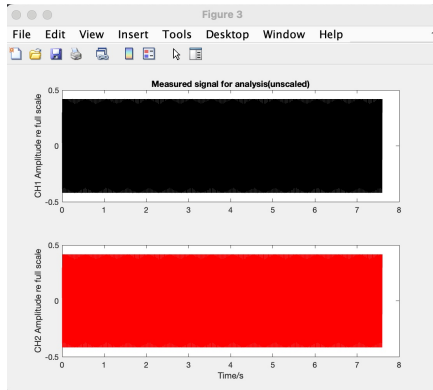
3 Calibration

Following the initial measurement, a calibration step is required. The first calibration focuses on calibrating the output of the Universal Analog Interface (UAI). This calibration, known as the sensitivity calibration of the output channel, allows us to accurately map the desired amplitude values to corresponding voltage levels.

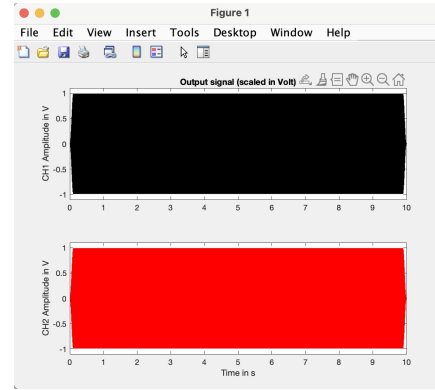
The second calibration is aimed at calibrating the input of the UAI. By calibrating the sensitivity of the input channel, we ensure precise voltage measurements. This calibration enables us to accurately determine the actual voltage levels of the input signal for more precise analysis and characterization.

3.1 Assignment 2

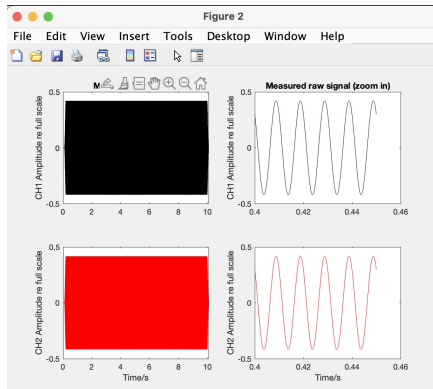
Upon executing the `calibration_single_sine.m` file, the graphs displayed in Figure 3 showcase the results.



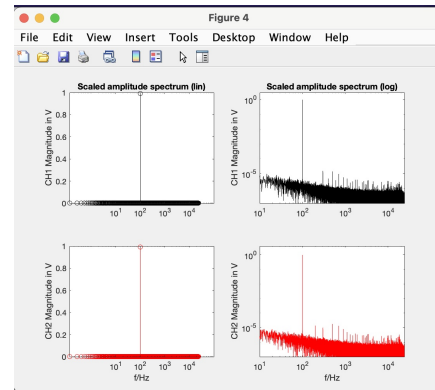
(a) Measured CH1 and CH2 Signal over time



(b) Output Signal



(c) Measured Signal with raw signal



(d) linear and logarithmic Scaled Amplitude Spectrum

Figure 3: Analysis of Measured Signals, Output Signal, and Amplitude Spectrum

To determine the output sensitivity and input sensitivity of the UAI, the provided MATLAB script `calibration_single_sine.m` uses the measured RMS voltage values at the UAI output and input, respectively. The unit of sensitivity is given in Volts per arbitrary unit $\frac{V}{a.u.}$.

3.1.1 Output Sensitivity Calculation

The output sensitivity is calculated by dividing the measured RMS voltage at the UAI output by the corresponding MATLAB-defined peak amplitude (1 a.u.). The formula is as follows:

$$\text{Output Sensitivity} = \frac{\text{RMS Voltage at UAI Output}}{\text{MATLAB Peak Amplitude (1 a.u.)}}$$

#MATLAB

```
sensitivity_out = amp ./ (rms_amp * sqrt(2));
```

3.1.2 Input Sensitivity Calculation

The input sensitivity is calculated by dividing the measured RMS voltage at the UAI input by the corresponding MATLAB-defined peak amplitude (1 a.u.). The formula is as follows:

$$\text{Input Sensitivity} = \frac{\text{RMS Voltage at UAI Input}}{\text{MATLAB Peak Amplitude (1 a.u.)}}$$

#MATLAB

```
sensitivity_in = rms_amp * sqrt(2) ./ a_f;
```

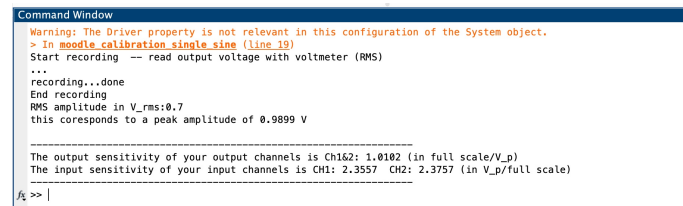
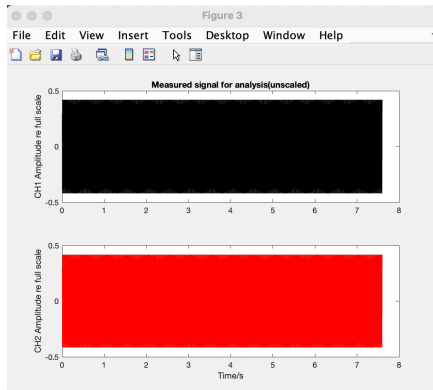


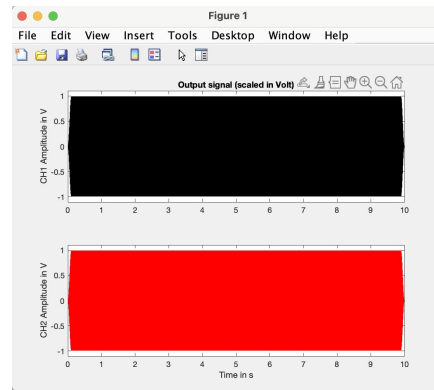
Figure 4: Results of Input and Output Sensitivity

To plot the result of Assignment 1a with scaled axes in physical units (V and s), the sensitivity values are used. The sensitivity values represent the conversion factor between digital units and physical units. By multiplying the digital values with the respective sensitivity values, the axes can be scaled to represent voltage in volts (V) and time in seconds (s). This scaling allows for a more meaningful representation of the data in the plot, providing a clearer understanding of the signal characteristics in physical units.

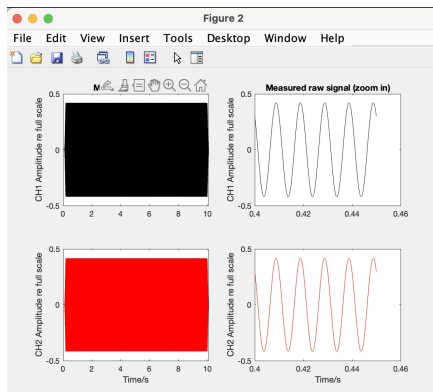
If RMS amplitude is 0.7, it correspond to a peak amplitude of 0.9899V, the output sensitivity of your output channels is 1.0102 for both Channels. For our case, the output sensitivity of the output channels is CH1 and CH2: 1.0102 (in full scale/ V_p) and the input sensitivity of the input channels is CH1: 1.3273 CH2: 1.3400 (in V_p /full scale)



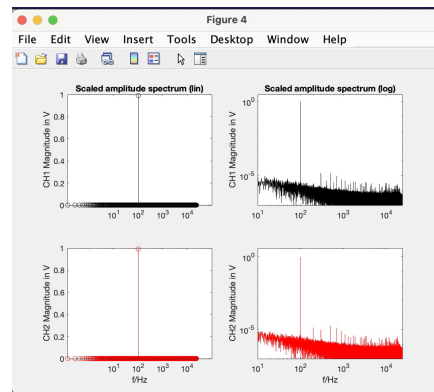
(a) Output Signal of playData



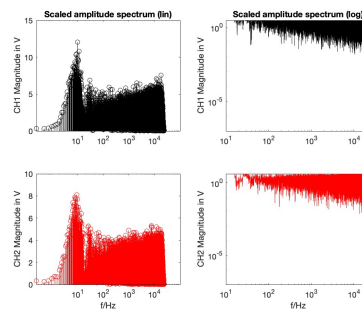
(b) Output Signal



(c) Measured Signal with raw signal



(d) unscaled Measured Signal



(e) linear and logarithmic scaled amplitude spectrum

Figure 5: Analysis of Measured Signals, Output Signal, and Amplitude Spectrum using sensitivity values

4 Latency and Jitter

Latency, in general, refers to a brief delay that occurs when a signal enters a system and then emerges after a certain period of time. This delay

Taeyoung Kim, SS2023

```
Command window
Start recording -- read output voltage with voltmeter (RMS)
...
recording...done
End recording
RMS amplitude in V_rms:0.7
this corresponds to a peak amplitude of 0.9899 V

-----
The output sensitivity of your output channels is Ch162: 1.0102 (in full scale/V_p)
The input sensitivity of your input channels is CH1: 7814.1203 CH2: 6167.3915 (in V_p/full scale)
fg >>
```

Figure 6: Value of Output and Input sensitivity

is typically measured in milliseconds and can have significant implications for the performance of professional measurement systems. Latency is a crucial performance metric as it directly impacts the timeliness and accuracy of signal processing and transmission. Ensuring low latency is essential in applications where real-time data processing and synchronization are vital for optimal system performance and user experience.

4.1 Assignment 3

Latency, in general, refers to a brief delay that occurs when a signal enters a system and then emerges after a certain period of time. This delay is typically measured in milliseconds and can have significant implications for the performance of professional measurement systems. Latency is a crucial performance metric as it directly impacts the timeliness and accuracy of signal processing and transmission. Ensuring low latency is essential in applications where real-time data processing and synchronization are vital for optimal system performance and user experience.

4.1.1 Latency

In this assignment, the goal is to measure the round-trip delay of the system. This is achieved by generating an impulse signal vector, representing a Kronecker delta of 1 arbitrary unit (a.u.), on both output channels of the system. The impulse signal vector is created by initializing a 250 ms long zero vector using the zeros function and then setting a single sample at 20 ms to 1. This process generates a brief pulse in the signal, which will be used to analyze the round-trip delay in subsequent steps of the assignment.

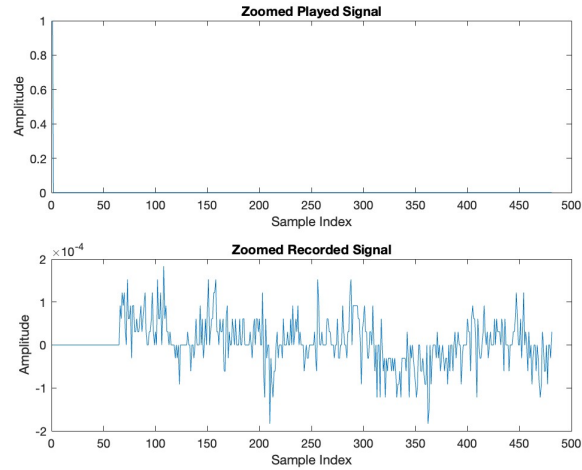
```
#MATLAB
%% Generate impulse signal vector
% Calculate the number of samples
numSamples = round(T * fs);

% Create a zero vector
impulseSignal = zeros(numSamples, 1);
```

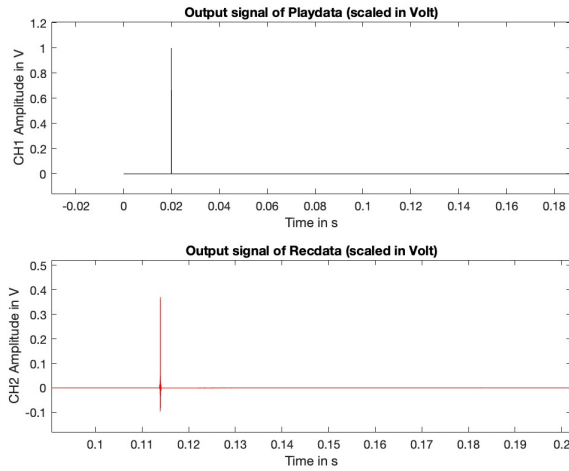
```

% Set the sample at 20 ms to 1
sampleIndex = round(0.02 * fs); % 20 ms
impulseSignal(sampleIndex) = 1;

```



(a) Zoomed played and recorded signal



(b) Output Signal of played and recorded signal

Figure 7: Result of played and recorded Output signal

The recorded signals may not appear as a simple Kronecker delta anymore due to various factors, including the characteristics of the system and the limitations of the hardware. These factors can introduce noise, distortion, and latency into the recorded signals. Additionally, factors such as analog-to-digital conversion, buffering, and other signal processing stages can also

contribute to the alteration of the recorded signals. Thus, the recorded signals may deviate from a perfect Kronecker delta, resulting in a more complex waveform.

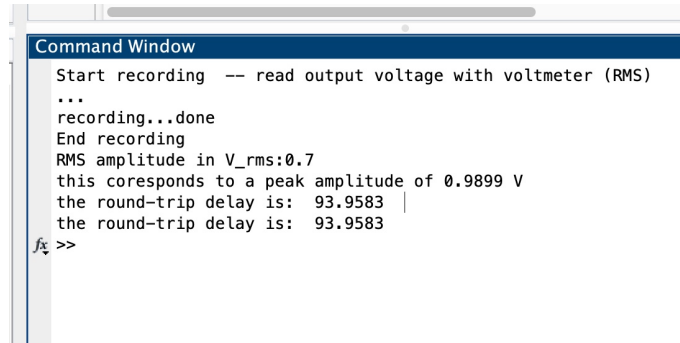
The round-trip delay of the system can be determined by finding the index of the maximum value in the recorded signal vector using the `max` function. By subtracting the index of the impulse from the index of the maximum value, we can calculate the round-trip delay in terms of the number of samples. To convert this delay to milliseconds, we divide the number of samples by the sampling rate and multiply by 1000. By performing these calculations with the appropriate values, we can determine the round-trip delay of the system in both samples and milliseconds. Also the round-trip delay value is 93.9583 in this case.

```
#MATLAB
% Find the index of the maximum value in the recorded signal
[~, maxIndex] = max(recData);

% Calculate the round-trip delay in number of samples
roundTripDelaySamples = maxIndex - sampleIndex;

% Calculate the round-trip delay in milliseconds
roundTripDelayMilliseconds = roundTripDelaySamples / fs * 1000;

fprintf('the round-trip delay is: %.4f \n',roundTripDelayMilliseconds)
```

A screenshot of the MATLAB Command Window. The window has a blue title bar that says "Command Window". The text inside the window shows the execution of a script: "Start recording -- read output voltage with voltmeter (RMS)", followed by three dots "...", then "recording...done", "End recording", "RMS amplitude in V_rms:0.7", "this corresponds to a peak amplitude of 0.9899 V", "the round-trip delay is: 93.9583", and "the round-trip delay is: 93.9583". At the bottom, the prompt "fx >>" is visible.

```
Command Window
Start recording -- read output voltage with voltmeter (RMS)
...
recording...done
End recording
RMS amplitude in V_rms:0.7
this corresponds to a peak amplitude of 0.9899 V
the round-trip delay is: 93.9583
the round-trip delay is: 93.9583
fx >>
```

Figure 8: Results of round-trip delay

4.1.2 the principle of cross-correlation to determine the round-trip delay and potential jitter of the UAI

First generate a $T = 0.5$ s long, zero mean white noise signal with a peak amplitude of 0.5V. Using the ramping function ramp the signal on and off

over a duration of 100ms.

```
#MATLAB
% Compute the cross-correlation between the output and input signals
[correlation, lags] = xcorr(recData(:,1), input_signal(:,1));
```

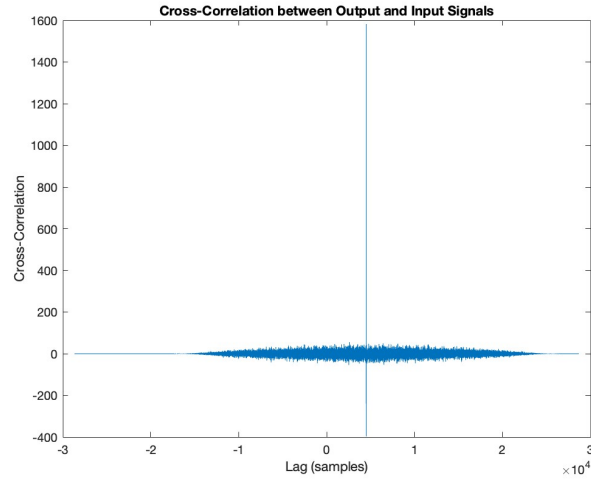


Figure 9: cross correlation between the output and the corresponding input signal

Repeat it ten times and calculate the round-trip delays and corresponding jitter value. The values are as follows:

Iteration	Round-trip delays (ms)
1	323.3542
2	263.7083
3	362.8958
4	442.5625
5	352.0208
6	357.2083
7	322.4583
8	362.7083
9	178.1667
10	225.7917

Table 1: A table with ten round-trip delay values

The potential jitter can be calculated by subtracting the max and min value of the round-trip delay. The obtained potential jitter value is 264.3958 ms.

```

Command Window
the round-trip delay is: 352.0208
Start recording -- read output voltage with voltmeter (RMS)
...
recording...done
End recording
this corresponds to a peak amplitude of 0.9899 V
the round-trip delay is: 357.2083
Start recording -- read output voltage with voltmeter (RMS)
...
recording...done
End recording
this corresponds to a peak amplitude of 0.9899 V
the round-trip delay is: 322.4583
Start recording -- read output voltage with voltmeter (RMS)
...
recording...done
End recording
this corresponds to a peak amplitude of 0.9899 V
the round-trip delay is: 362.7083
Start recording -- read output voltage with voltmeter (RMS)
...
recording...done
End recording
this corresponds to a peak amplitude of 0.9899 V
the round-trip delay is: 178.1667
Start recording -- read output voltage with voltmeter (RMS)
...
recording...done
End recording
this corresponds to a peak amplitude of 0.9899 V
the round-trip delay is: 225.7917
Round-Trip Delays:
323.3542 263.7083 362.8958 442.5625 352.0208 357.2083 322.4583 362.7083 178.1667 225.7917
Jitter:
264.3958

```

Figure 10: Results of round-trip delay repeated ten times

The first technique involves using an input signal as an impulse or Kro-necker delta. The round-trip delay is determined by analyzing the recorded output signal to find the corresponding time delay.

The second technique involves using a white noise input signal and applying the cross-correlation function to calculate the round-trip delay. This technique relies on finding the correlation between the output and input signals to identify the delay.

It is important to note that the specific results and observations regarding the differences in round-trip delay between the two techniques would depend on the actual measurements performed and the analysis conducted.