Boston University
Electrical & Computer Engineering
EC463 Senior Design Project

Prototype Test Report

On-Board-Sound-Intensity Data Acquisition System (OBSIDAS)

by

Team 8
Volpe

Team Members

Muhamed Abdalla muhabda@bu.edu
Aurojit Chakraborty aurojit@bu.edu
Daniel Cardosi dcardosi@bu.edu
Javier Gonzalez Santamaria javiergs@bu.edu
Joseph Lucatorto III jlucator@bu.edu

November 14, 2023

1. Required Materials

Hardware:

- 4 GRAS 26AK preamplifiers (serial numbers 96954, 96952, 96948, 96951)
- 2 GRAS 40AI intensity-microphone pairs (serial numbers 80454 and 80490 & 80433 and 80441)
- 1 Bruel and Kjaer 4231 calibrator (serial number 1882979)
- 2 GRAS 12AQ 2-channel power modules w/ associated batteries and power cables (serial numbers 367569, 354702)
- 4 GRAS 7-pin LEMO 10-meter extension cables
- 4 BNC plug 25-ft extension cables
- Personal computer (for software testing)

Software:

- The Fast Fourier Transform (FFT) feature of the oscilloscope
- MATLAB R2023a (with testing program)

2. Set up

The purpose of this test was to ensure both the functionality of the hardware provided by our client and the validity of the computational methods to be implemented in our final program. All aspects of the set-up followed the previously written test plan.

The hardware was connected in a manner simulating that of the final measurement setup with which our software will be utilized. Each individual GRAS 40AI microphone, prior to testing, was first oriented with its capsule within the Bruel and Kjaer 4231 calibrator and subsequently connected to the lab's oscilloscope as follows: the probe was attached to one of the GRAS 26AK preamplifiers, and this preamplifier was then connected to an input channel on one of the GRAS 12AQ power modules via a 7-pin LEMO cable. The calibrator was then turned on, generating a 1-kHz tone at 94 dB SPL (at the driver's surface, against which the microphone's capsule was placed), and the oscilloscope was configured to display the FFT-computed spectrum of the input from the microphone. The power module was configured with a gain of 0 dB, and its "Filter" knob was set to "Lin" (for 'linear'). Finally, the power module was connected to the oscilloscope via one of the BNC coaxial cables.

The computational methods were tested via a MATLAB program mirroring the structure and content of the C++ script to be called by our final LabVIEW program. The only preliminary step was to run a script in order to load all Volpe-provided .wav files into MATLAB as real-valued numeric vectors, of which there were eight in total (being two signals, a calibration signal and a tire-pavement-noise signal, as recorded by each of the four microphones: the inside microphone of the leading probe, the outside microphone of the leading probe, the inside microphone of the trailing probe, and the outside microphone of the trailing probe).

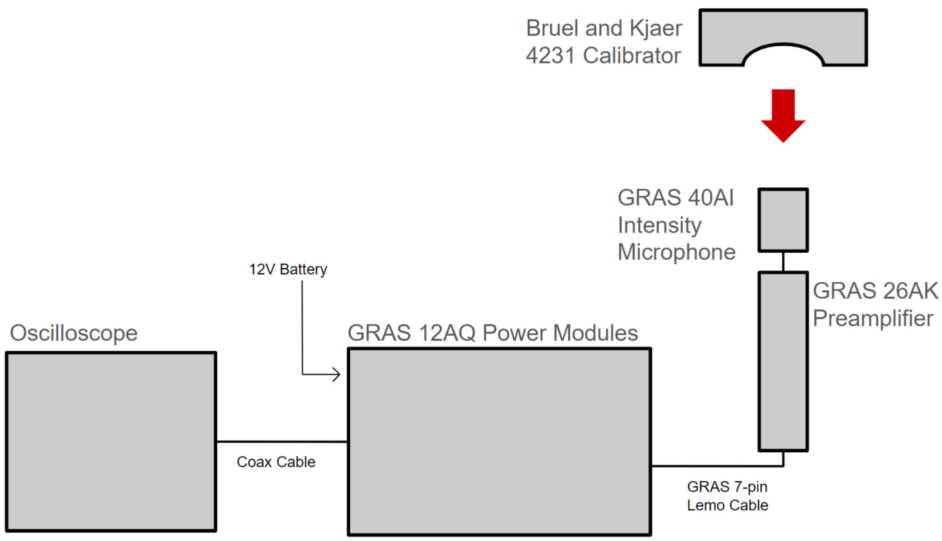

Fig. 1. Schematized diagram of hardware setup.

3. Testing Procedure:

After completing the described setup, the testing steps for the hardware were as follows:

1. Turn on the calibrator.
2. Switch on the GRAS 12AQ power module and set it to 0dB gain and 'Lin' for linear response, with a high-pass filter of 0.2 Hz for the channel to which the microphone is connected.
3. Set up the oscilloscope so that the sound signal can be viewed with multiple periods. A vertical scale separation of 500 mV is appropriate.
4. Compute the FFT to display a spectral representation of the test signal. If necessary, adjust the horizontal center position to display the 1kHz peak. .

The steps for testing the MATLAB program were as follows.

1. The spectrograms for each of the eight signals were displayed in order to verify their proper loading into MATLAB as one-dimensional real-valued sequences.
2. The DFT magnitudes across frequency were displayed for each of the eight signals.
3. The unscaled and scaled dB SPL values (for the DFT bin corresponding to 1 kHz) were displayed for the calibration signal for each of the four microphones to verify the sensitivity-parameter calibration script.
4. The ⅓-octave-band SPL values for both the calibration and noise signals were displayed for each of the four microphones.
5. The full-spectrum inter-microphone coherence values for the noise signals for each of the two probes were displayed, along with the corresponding slot-reversed values and the coherences of one of each signal type with itself.

4. Measurable Criteria

The criteria for the hardware input chain are as follows:

1. The microphone should successfully capture the sound from the calibrator.

2. The power module should successfully amplify the signal.

3. The signal should be successfully displayed on the oscilloscope as a range of amplitude values across frequency, with a peak in the spectrum at 1 kHz.

The criteria for the MATLAB program were as follows:

1. The spectrograms for the calibration signals should show a concentration of energy at 1 kHz, and those for the noise signals should show a more broadband distribution of energy across the recorded spectrum.
2. The DFT-magnitude plot in all cases should show Hermitian symmetry, as the input sequences are real-valued.
3. The scaled 1-kHz-bin SPL values for the calibration signal should be 94 dB for each of the four microphones, regardless of the unscaled values.
4. Each of the single-band SPL values should be within the dynamic range of the GRAS 40AI, between 20 dB and 152 dB.
5. Each of the outputted coherence values should be between 0 and 1. The normal and slot-reversed computations should return identical values, and the self-coherences should be exactly 1.

5. Results

All aforementioned criteria were met for both the hardware- and software-related components of the testing.

For the hardware tests, Fig. 2 illustrates the visualization of the calibrator's sound signal on the oscilloscope, confirming that the microphone captured the calibration signal. After computing the FFT to display a spectral representation of the test signal, it was necessary to adjust the horizontal center position to view the 1-kHz peak. The signal illustration in Fig. 3 reveals this peak at 1 kHz, indicating a successful test. Repeating this procedure for all microphones, preamplifiers, power-module input channels, and associated cables revealed no improper operation by any component, confirming that all hardware performs as expected.

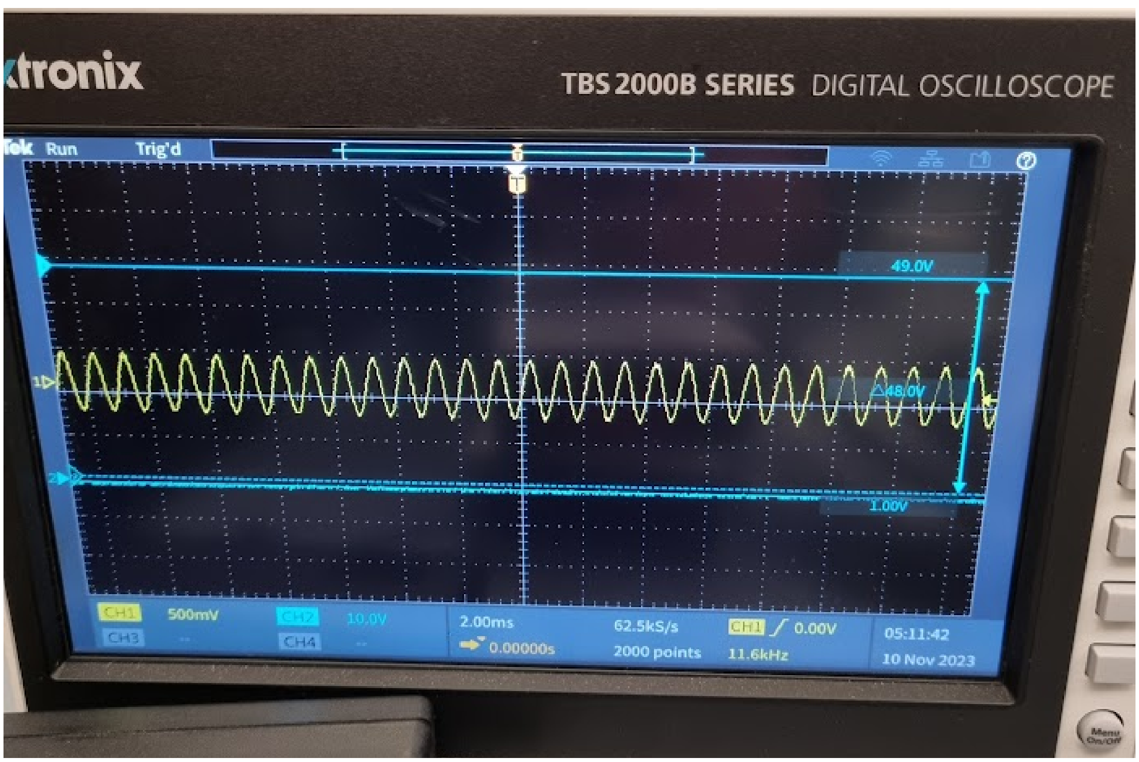

Fig. 2. Sound signal captured by microphone, displayed on the oscilloscope as a plot of amplitude vs. time.

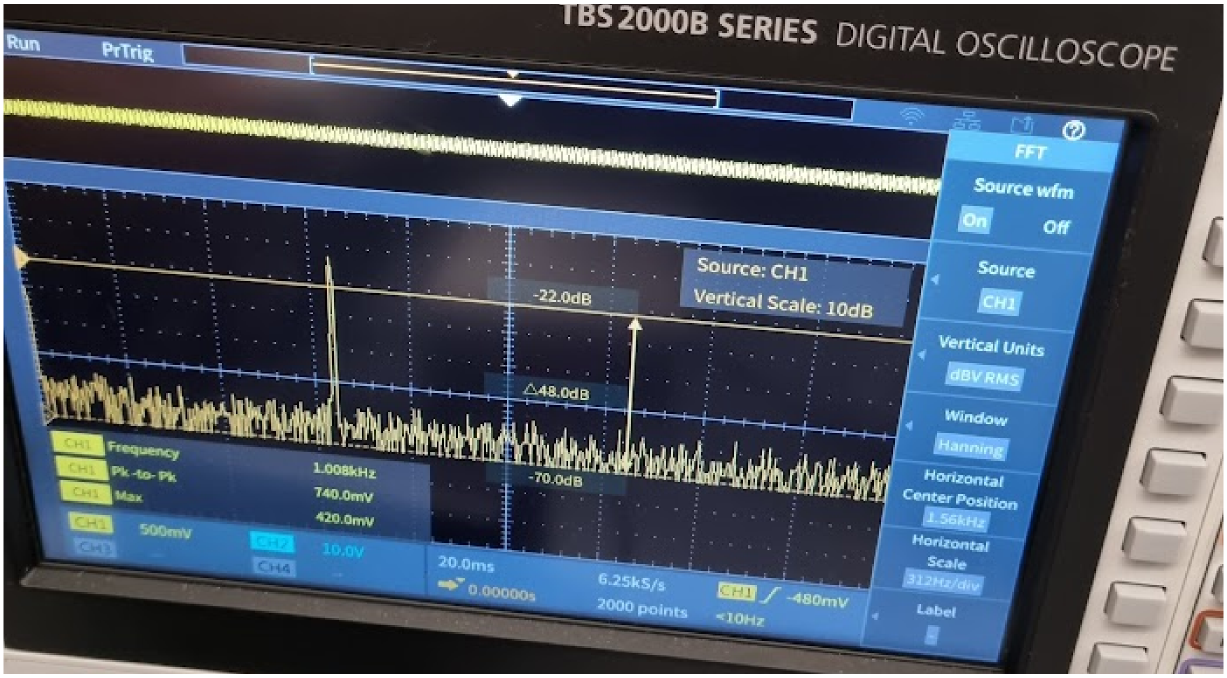

Fig. 3. FFT of sound signal captured by microphone.

For the software tests, appearing below are images of representative outputs at the first two stages described above. As visible in Fig. 4, the spectrograms generated for the calibration tones showed in all cases a concentration of energy at 1 kHz throughout the duration of the signal, and those generated for the noise signals showed no such concentration, confirming the correct conversion of the .wav files into numeric sequences. By request of the graders, an example of the latter was also played through the computer's built-in speakers via MATLAB's sound() function in order to verify in particular the high relative energy in the low-frequency region of the spectrum, and as expected, the noise resembled that of a muffled rumble, audibly having comparatively little information above 1 kHz. As visible in Fig. 5, the DFT-magnitude plots showed the same spectral differences, in all cases also displaying the expected Hermitian symmetry and thus confirming the functionality of the FFTW library's unshifted-DFT algorithm, to be utilized in the final C++ script (the specific routine in C++ is defined by the fftw_plan object "fftw_plan_dft_r2c_1d"). Fig. 6 tabulates the unscaled and scaled 1-kHz-bin SPL values for all four microphones, in each case confirming the functionality of the sensitivity-scaling script, and Fig. 7 confirms that after scaling, the SPL-computation script returns no values for any of the ⅓-octave bands in any of the microphones that fall outside the range of possible values determined by the 40AI's dynamic range. As visible in Fig. 8, lastly, all computed inter-microphone coherences had values between 0 and 1; slot-reversal of sequence pairs yielded no changes in values, confirming the conjugate-symmetry (in this case extended to ordinary symmetry, given the real-valued inputs) of the operator as implemented into code; and the computed self-coherences were all equal to 1, confirming the definitional (and intuitive) fact that any signal is 100% coherent with itself.

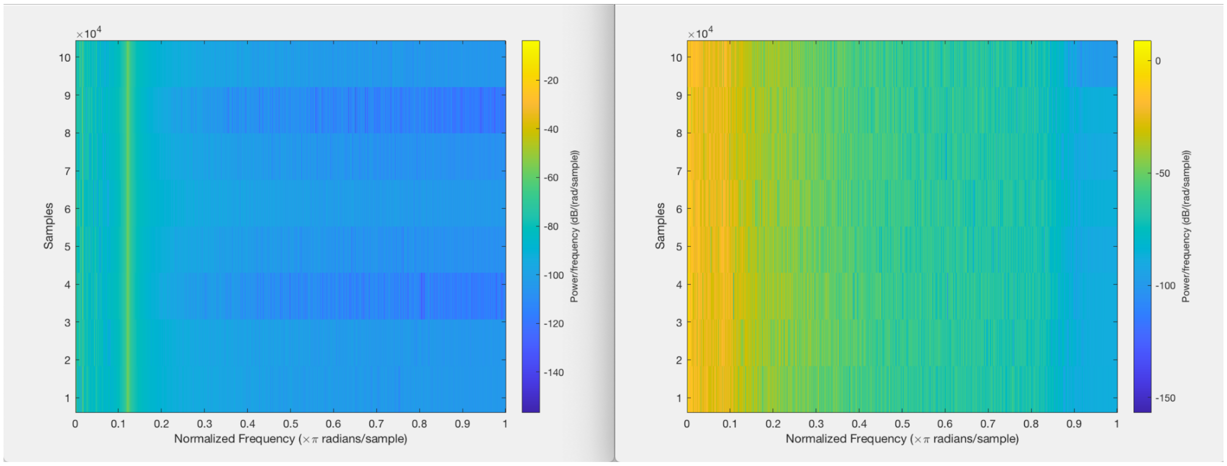

Fig. 4. Spectrograms of calibration signal (left) and tire-pavement-noise signal (right) as recorded by the inside microphone of the leading probe.

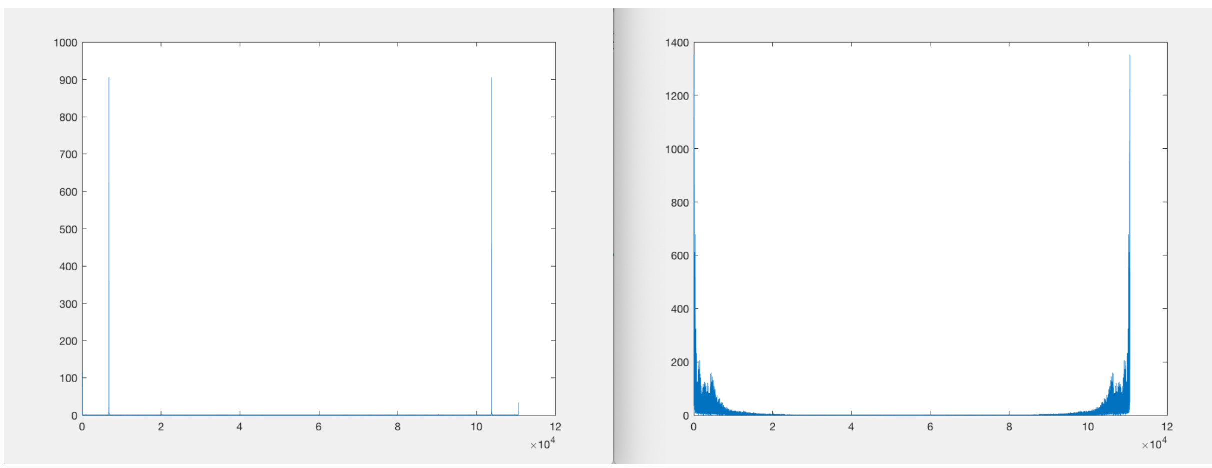

Fig. 5. DFT-magnitude vs frequency plots for calibration signal (left) and tire-pavement-noise signal (right) as recorded by the inside microphone of the leading probe.

	dB, pre-scaling	dB, post-scaling
Leading Probe, Inside Mic	30.1853	94.0000
Leading Probe, Outside Mic	30.7402	94.0000
Trailing Probe, Inside Mic	30.0662	94.0000
Trailing Probe, Outside Mic	29.3313	94.0000

Fig. 6. Unscaled and scaled dB(SPL) values for calibration signal as recorded by each microphone.

	Leading Probe, Inside Mic	Leading Probe, Outside Mic	Trailing Probe, Inside Mic	Trailing Probe, Outside Mic
dB, 250 Hz	83.6553	82.1948	82.3950	82.3997
dB, 315 Hz	80.9772	79.1851	79.4781	79.1428
dB, 396.9 Hz	82.2006	80.6066	80.5167	80.3949
dB, 500 Hz	80.8394	79.2949	79.9006	79.8471
dB, 630 Hz	82.7749	81.2477	79.7366	79.7342
dB, 793.7 Hz	80.2573	78.7950	80.3184	80.3181
dB, 1000 Hz	74.5697	72.9488	75.8519	75.6874
dB, 1259.9 Hz	69.0080	67.3399	70.9140	70.7598
dB, 1587.4 Hz	64.4980	62.8084	64.6538	64.4377
dB, 2000 Hz	61.1505	59.4776	59.8347	59.6255
dB, 2519.8 Hz	56.2649	54.6486	56.5263	56.3640
dB, 3174.8 Hz	51.1096	49.9006	52.1326	52.1920
dB, 4000 Hz	45.3986	44.0372	46.5788	46.4683
dB, 5039.7 Hz	41.4956	40.1785	42.1175	41.6894

Fig. 7. Scaled dB(SPL) values for each ⅓-octave band (band-center frequencies in leftmost column), for noise signal as recorded by each microphone.

	Coherence	Coherence, slot-reversed computation
Leading probe, both microphones	0.7993	0.7993
Leading probe, inside mic (in both slots)	1.0000	-----
Trailing probe, both microphones	0.3182	0.3182
Trailing probe, inside mic (in both slots)	1.0000	-----

Fig. 8. Inter-microphone coherence values (unitless) for tire-pavement-noise signal.

6. Conclusions

The testing described herein for the On-Board-Sound-Intensity Data Acquisition System (OBSIDAS) prototype was completely successful, validating the functionality of both the hardware provided by our client and the software components thus far developed.

All client-provided hardware (GRAS preamplifiers, intensity-microphone pairs, requisite cables, and a calibrator) with the exception of the National Instruments USB-4431 data acquisition system were included in the test; as the USB-4431 was confirmed at an earlier date to interface properly (as an ADC) with a personal computer, the tests detailed above served to examine the signal chain of our final system strictly up to the point of the analog-to-digital conversion preceding each input signal's processing within LabVIEW and C++. Oscilloscope visualizations (see Fig. 2) confirmed the accurate capture of the calibrator's sound signal by each microphone, additionally verifying the functionality of all preamplifiers and power-module input channels, and subsequent FFT analysis revealed in each case an expected peak in amplitude at 1 kHz, meeting the predefined criteria. The fourfold input chain, from signal generation by the calibrator, to signal capture by the microphones, to amplification by the preamplifiers, and finally to an external display as outputted by the power module, thus successfully fulfilled all measurable criteria, ensuring the robustness of the OBSIDAS hardware.

All tested software elements were likewise confirmed to perform properly: the subroutines for DFT computation (via the FFTW library), sensitivity-scaling (via a fixed-level 1 kHz tone), ⅓-octave-band decomposition, and inter-microphone-coherence computation operated as expected when applied to the Volpe-provided example .wav files. Each of these routines was copied into MATLAB directly from a preliminary version of the final C++ script, using only rudimentary mathematical functions (with the exception of the functions of the FFTW library) and statement types available in both C++ and MATLAB; accordingly, the functionality of these routines in MATLAB indicates their functionality in the developed C++ program.