ARL-TR-9053 ● Sep 2020

Night Vision Acoustic Array (NVAA)

by Tung-Duong Tran-Luu

Approved for public release; distribution is unlimited.

NOTICES

Disclaimers

ARL-TR-9053 ● SEP 2020

Night Vision Acoustic Array (NVAA)

Tung-Duong Tran-Luu
Computational and Information Sciences Directorate, CCDC Army Research Laboratory

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE			*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED (From - To)**
September 2020	Technical Report	1 February 2019–30 September 2019

4. TITLE AND SUBTITLE	**5a. CONTRACT NUMBER**
Night Vision Acoustic Array (NVAA)	
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	**5d. PROJECT NUMBER**
Tung-Duong Tran-Luu	
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	**8. PERFORMING ORGANIZATION REPORT NUMBER**
CCDC Army Research Laboratory ATTN: FCDD-RLC-CA Adelphi, MD 20783-1138	ARL-TR-9053
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

ORCID ID(s): Tung-Duong Tran-Luu, 0000-0001-6769-2142

14. ABSTRACT

This report describes the concept and design of a handheld acoustic array developed for the Night Vision and Electronic Sensors Directorate—the Night Vision Acoustic Array (NVAA)—that keeps track of the location of sounds detected in a battlefield environment. The NVAA's purpose is to alert a forward observer (FO) of gunfire, explosions, and other transient signals outside of their field of view. The targets are displayed on the small screen relative to the array pointing direction. The FO slews the array in each detected direction to inspect the target with the scope mounted on top of the case. The technical specifications and design parameters are described, along with some caveats given the nature of acoustic propagation. The signal processing features a novel coherence detector, which uses simultaneously data from all channels, unlike the conventional approach of averaging pairs of them. The theory for this new coherence is described, along with an illustration of the performance improvement over the conventional approach. A user's manual is also provided.

15. SUBJECT TERMS

acoustic array, coherence detection, source localization

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Tung-Duong Tran-Luu
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER (Include area code)**
Unclassified	Unclassified	Unclassified	UU	31	(301) 394-3082

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

1. Introduction

For the past three decades, the Army Research Laboratory (now the US Army Combat Capabilities Development Command [CCDC] Army Research Laboratory [ARL]) has been developing acoustic systems to locate targets such as tanks, helicopters, artillery, mortars, and small arms in the battlefield environment. Acoustic sensors offer many advantages: they are generally passive, omnidirectional, low power, compact, and low cost.

An important development happened in 2004 when the Unattended Transient Acoustic Measurement and Signature Intelligence (MASINT) System (UTAMS) was built and rushed to Iraq to counter mortar fire threats.[1,2] The UTAMS consists of four arrays placed at the corners of a square (1 km per side, roughly), with each array being a four-microphone tetrahedron of 1-m radius (Fig. 1). Each array produces a line of bearing (LOB) to the target, which is sent to a central computer node. The latter intersects the received LOBs to produce the location of the target. The UTAMS successfully detected and accurately localized many mortar fires, rocket fires, and improvised explosive device explosions around military bases. In 2009, the UTAMS was incorporated as a cueing system for the Persistent Threat Detection System aerostats, a program run by the Program Manager Robotics and Unmanned Systems.

Fig. 1 UTAMS array

In FY19, the CCDC Army Research Laboratory developed the Night Vision Acoustic Array (NVAA) system (Fig. 2) for the Night Vision and Electronic Sensors Directorate (NVSED).[3] The NVAA is an eight-microphone array intended to be built into the next-generation, man-portable targeting system for forward observers (FOs). It is handheld, low power (less than 10 W), and similar in size to the Joint Effects Targeting System (JETS), 10 × 7.5 × 6 inches.

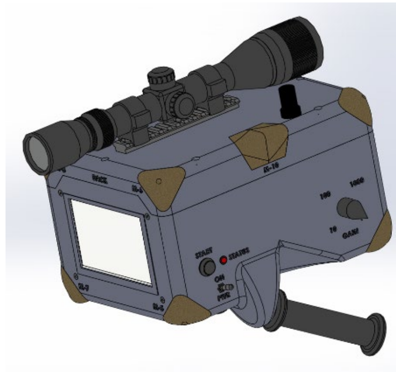

Fig. 2 NVAA unit

This technical report describes the NVAA, its purpose, concept of operations (CONOPS), requirements, specifications, signal processing, and algorithms. A user's manual is included as an Appendix.

1.1 Objective

To augment next-generation, man-portable targeting systems (a.k.a. the Multi-Domain User Sensor Architecture [MDUSA]) with acoustic capability, NVESD evaluated different standalone acoustic systems in FY19 and defined the specifications for integration with MDUSA in FY20. To this end, ARL has built a prototype array, the NVAA.

1.2 Operation

The expected CONOPS is a FO in a combat zone, possibly urban, having to locate optically targets such as indirect fires and small arms, and be alerted of new targets outside their field of view (FOV) by acoustics.

1.3 Requirements

The system is handheld, so it must be small, lightweight, low power, and similar in size to the JETS (10 × 7.5 × 6 inches).

The expected range is 300 m to the target at minimum, with typically 5 km in daytime and 2 km at nighttime.

Even though range is a useful military specification, it is hard to satisfy with a single passive acoustic sensor. Any passive detection boils down to the received signal-to-noise ratio, which is not a reliable indicator of distance to source, even if the latter is known completely. The reason is that the propagation channel can strongly affect the signal, even severely suppress it, as in the case of upward refraction, or make the signal louder than it would normally be, as in the case of ducting during temperature inversion (the sound is trapped between a cooler ground layer and a warmer upper air layer). Knowing that, the propagation channel would require measuring parameters such as wind profile as function of height, temperature profile, terrain, manmade structures, random air turbulences, and so on, many of which are impractical to measure over the propagation path at all times.

Multiple acoustic arrays networked together can work out the location of (hence distance to) the target. However, this approach does not fit the current CONOPS. Thus, this system does not inform the user about range to detected target, only the direction in 3-D space.

2. Specifications

2.1 Features

The following are the features of the NVAA:

- Eight-microphone array, configurable on-the-fly
- Automatic target detection and reporting
    - Slews detection dots (maximum of three) on the screen as the operator moves the device.
    - Detection dots are color-coded based on direction (front = blue, back = green) and confidence (dark = confident, light = not so much).
- Manually adjustable gain (1, 10, 100, and 1000 times) for very small or very large signals
- Handles to guarantee that the hands of the user will be at a predictable place, away from the microphones
- Thermometer mounted to avoid direct sunlight and body heat from the operator
- Heat sink plate to cool electronics inside the case
- Central processing unit (CPU) usage (v1.0)
    - 65% at 2 kHz, 75% at 2.5 kHz (+15% for graphical interface, v0.74)

2.2 System

The following are the system details:

- Microphone array
  - 10 microphones, 8 active at most at any time
  - Sampling rate at 2.5 kHz
- Case size (without handles or scope)
  - 10 L × 7.5 W × 6 H (inches)
  - 3-D printed
  - Schematics are shown in Figs. 3 and 4.
- Target Sporting Scope 3-9x40R, mounted on a Picatinny rail
- Power
  - Lithium-ion battery BB-2590U
  - 0.8 A at 12 V (measured)

NVESD Array Design3

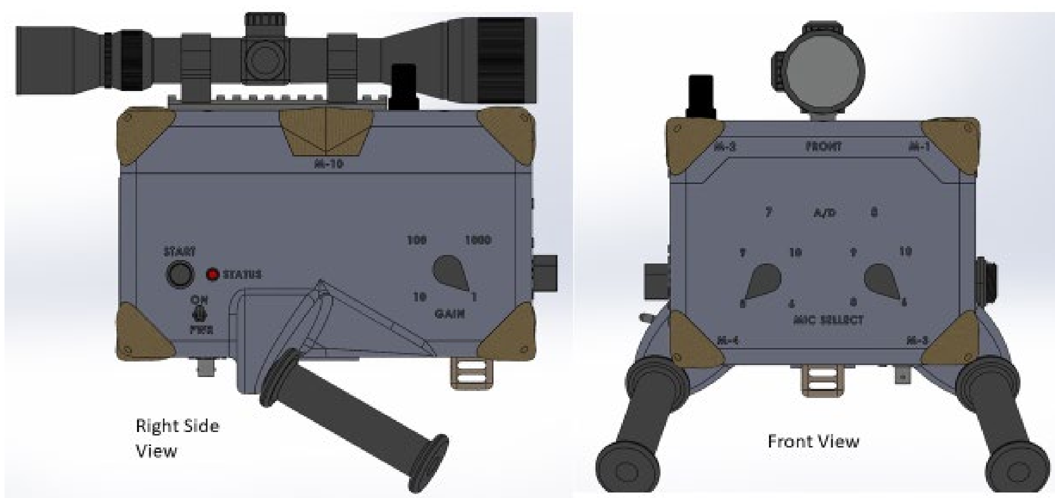

Fig. 3 Right and front view

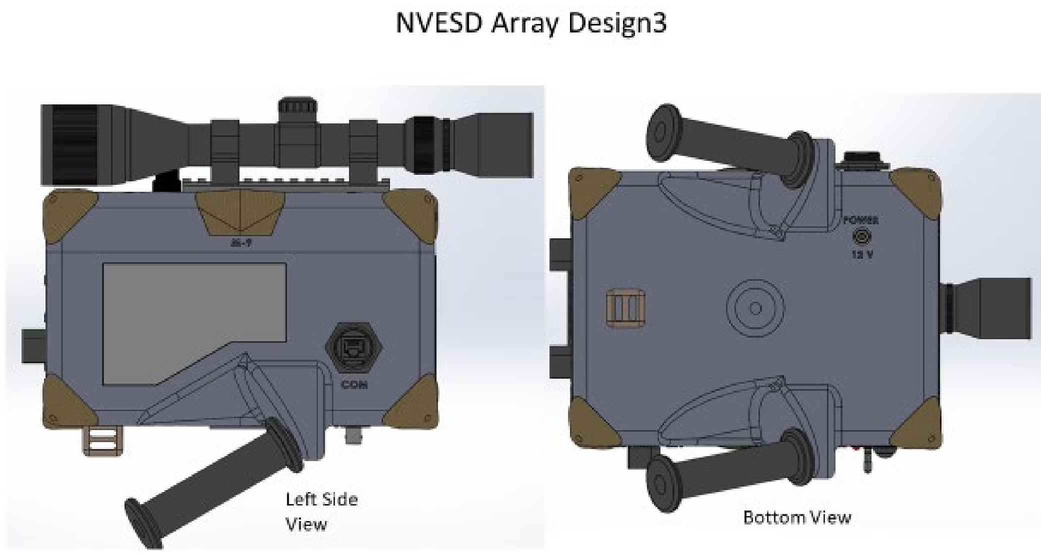

Fig. 4 Left and bottom view

2.3 Components

The following are the components of the NVAA:

- A 9-degrees-of-freedom inertial measurement unit (IMU) for accelerometer, gyroscope, and magnetometer (integrated with the Xilinx board)
- GPS (integrated with the Xilinx board), only used for log file time
- Temperature sensor
  - Maxim Integrated DS18B20
  - Measures temperatures from –55 °C to +125 °C (–67 °F to +257 °F)
  - ±0.5 °C accuracy from –10 °C to +85 °C
  - Programmable resolution from 9 to 12 bits
  - 1-wire bus, no external components required

2.4 Microphones

The following are the details of the microphones:

- Knowles VEK-H-30230-000
  - Sensitivity ±3 dB from 100 Hz to 10 kHz
  - Maximum dynamic range is 124 dB SPL

- Phase matched to at least within ±5° maximum, or ±1° mean, from 20 Hz to 2 kHz.

2.5 Analog-to-Digital Converter (ADC)

The following are the details of the ADC:

- 8-channel 24-bit ADC
  - Frequency response: 6 Hz to 6.7 KHz (3 dB), third order
  - System dynamic range: 120 dB, limited by a 24-bit ADC
  - GPS timestamp: less than 1-μs accuracy (in this implementation, the data are timestamped by the CPU*)
  - Signal conditioning unit
  - At 16-kHz sampling, one gets 3 bytes per sample including header overhead. The real-time storage rate is 0.5 MB/s (2 GB/h).

2.6 Processor

The following are the details of the processor:

- System board
  - snickerdoodle, from krtkl
- Xilinx Zynq-7010 system-on-chip
  - Dual 32-bit 100- to 800-MHz ARM Cortex-A9 processors
  - Xilinx Artix-7 field-programmable gate array
  - 1-GB low-power mobile DDR2 random access memory
  - 16-MB NOR flash
  - Bluetooth 4.1, 150 Mbps 2.4 GHz, 802.11n WiFi with selectable internal or external antenna
  - MicroSD card, maximum 200-GB system storage and firmware update

*Due to excessive ringing due to signal processing in the field-programmable gate array, the data are taken directly from the ADC. It is therefore not timestamped.

2.7 Display

The following are the details of the display:

- Matrix Orbital GTT50A
  - Color 5-inch thin-film transistor LED resistive touchscreen 800 × 480 (WVGA)
  - 500 mA at 5 V (typical)

3. Signal Processing

The operating system running on the snickerdoodle is Ubuntu 16.04 Long Term Support. The software for signal processing and user interface is written in Python 3.6, with critical sections in Cython, version 0.23.4. Cython is more or less C with Python syntax.

3.1 General Architecture

Three processes and one thread run in parallel and communicate through queues or pipe:

1) Data input
   a) `a2d_get_stream()`, process in Python
   b) Read the data packet (160 samples × 8 channels) from the driver.
   c) Form the packet header and reformat the data.
   d) Read the IMU data.
   e) Write the data to the pipe and the IMU data to the global variable.
2) Data processing
   a) `sip.pyx()`, process in Cython
   b) Read the data packets from the pipe and process them (details in Section 3.2).
   c) Write the results (time of detection, angles of detection, etc.) to the result queues.
3) Data output
   a) `log_output.py()`, process in Python

b) Read the detection results from the result queues.

c) Write the text results to the log files.

4) User input

a) `user_input.py()`, thread in Python

b) Read the input from the touchscreen and detection results from the result queues.

c) Execute user commands from the graphical user interface and update the screen with dots or text messages.

3.2 Data Processing

3.2.1 Preprocessing

The signal read out from the Python process is high-passed to remove the DC component and passed through an adaptive filter to remove the background noise. The stopband of the high-pass filter is adjusted automatically to reject the low frequencies that produce trends in the analysis windows (the Fourier data windows). In this design, the adaptive filter will remove continuous-wave signals, such as engines or air-conditioner units running in the background.

1) High-pass filter to remove the DC component

a) Chebyshev type II filter (selected because it has no ripple in the pass band)

b) high pass from 20 Hz (stop band, tentative) to 40 Hz (pass band)

c) It loses no more than 2 dB in the pass band and has at least 60-dB attenuation in the stop band.

d) The stop band is increased to match or exceed the fundamental frequency of the fast Fourier transform (FFT) window length (32 samples).

2) Linear predictive filter to remove background noise

a) Window length: 16 samples

b) Coefficients updated with a Kalman filter (discussed in more detail in the following sections)

3.2.2 Adaptive Filtering

1) At startup, noise data are accumulated for the (number of microphones × buffer length) samples.

2) The buffer length cannot be too short, because the cross-correlation windows might be clipped by the left boundary. Now it is equal to 20 times the packet length.

3) Initial Kalman parameters

    a) The initial covariance matrix is the empirical covariance matrix of the noise data and the initial adaptive filter coefficients are solved from the Wiener–Hopf equation.

4) Every eight noise detection frames, the parameters are updated. Faster updates will require more CPU time.

3.2.3 Coherence Computation

After the signal has been cleaned, the detection feature is computed. If the signal is coherent across all data channels (e.g., they are shifted version of each other), then the output value will be near 1. Otherwise, it is low (near 0, if it were not for the background clutter, which might be coherent):

1) Short-time FFT, no padding

    a) Window length: 32 samples

    b) Step size: 32 samples

2) Coherence

    a) It uses four consecutive FFT windows, forming one detection frame.

    b) The step size is one FFT window.

    c) This coherence for multiple signals (>2) is novel and is discussed in detail in Section 3.3.

3) For each data packet, all detection frames that fully fit inside the packet get assigned a coherence value.as many detection frames are computed as possible, each frame getting one coherence value.

3.2.4 Detection

The detection used the coherence value computed for consecutive data frames to decide which frames have a signal of interest and which do not:

1) Each detection frame is tested in sequence (its coherence is compared to threshold = 0.8).

2) The loop remembers its previous state (had a detection or not) to gather a sequence of detected frames, building up the event signal.

3) There is no detection for the first 10 detection frames to let the adaptive filter adapt.

3.2.5 Signal Segmentation

After each event signal has been built, the leading edge is extracted out, while the tail, which might contains signal reflection, is truncated. Ten different truncations are attempted since there is no simple algorithm to determine where and if any reflection exists in the signal:

1) Once the event signal is built from the detected frames, 10 candidate sub-segments are used for computing the direction of arrival (DoA).

2) The purpose is to focus on leading edge and remove reflection, which is generally in the tail of the signal.

3) The segment is dropped if the estimated maximum delay is too large for the physical size of the array. It is also ignored if the estimated maximum delay does not exceed the delay across the two closest neighboring microphones.

4) This "filtering" cuts down on the CPU usage, which is more intense during DoA calculation. However, it might lead to missed detections when the time delay estimate is too noisy.

3.2.6 DoA Computation

The time delay between the data channels is estimated for each candidate sub-segment. The microphone furthest from the source is removed to avoid the diffraction effect of the housing. If the DoA least-squares fit error is still not small enough, the two microphones farthest from the source are removed:

1) The time difference (delay) between microphones is estimated from cross-correlating pairs of them.

2) The DoA is found by fitting those delays to a propagating plane wave model.

3) The linear least-squares fit is weighted by the peak of the cross-correlation.

4) Diffraction around the edges produces extra delays for the sensors in the (acoustic) shadow of case (Fig. 5). The issue is handled by removing the

time-delay values corresponding to the sensor with the highest delay (the one farthest from the source).

5) When the estimated speed of sound, c_est, is greater than 1.09 times the speed of sound derived from measured temperature, c_meas, one microphone is removed. When it is greater than 1.15 times c_meas, two microphones are removed.

6) If no solution has c_est between 0.5 and 2, the event is not reported.

7) Note that a coherent signal can still have a time-delay configuration that is incompatible with the array geometry. The reason is that coherence is frequency based, so reflection effects are ignored, while time delays are very sensitive to multipaths.

8) Dropping a valid event causes again a missed detection, but this software version (1.0) does not have an option for reporting an event that it cannot localize.

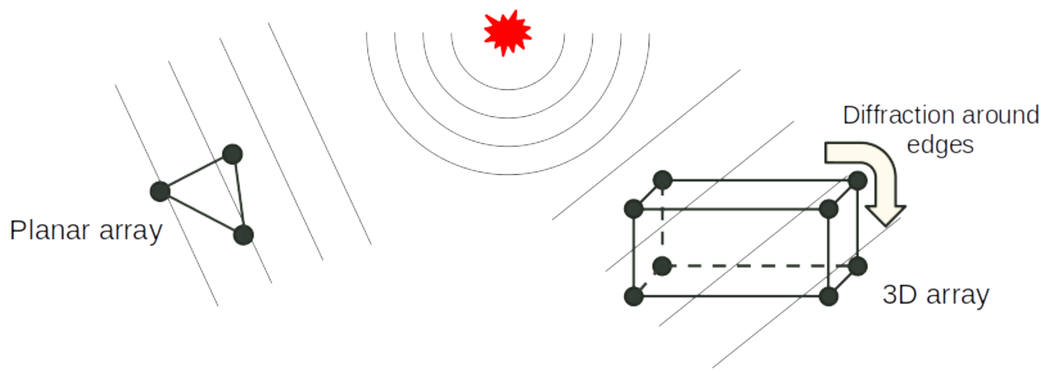

Fig. 5 Diffraction around the edges of the housing

3.3 Theory for Coherence Computation

The standard formula for coherence between two stationary random signals $x(t)$ and $y(t)$ is

$$C_{xy}(f) = \frac{|G_{xy}(f)|^2}{G_{xx}(f)G_{yy}(f)} \tag{1}$$

where $G_{xy}(f)$ is the cross-spectral density between $x(.)$ and $y(.)$, and $G_{xx}(f)$ and $G_{yy}(f)$ their respective auto-spectral densities.[4] For stationary signals, those quantities can be estimated by their time average:

$$\hat{G}_{xy}(f) = \frac{1}{N}\sum_{k=1}^{N} X(t_{k,}\, f)\, Y(t_k, f)^*, \tag{2}$$

$$\hat{G}_{xx}(f) = \frac{1}{N}\sum_{k=1}^{N} X(t_{k,} f)\, X(t_k, f)^* \tag{3}$$

and

$$\hat{G}_{yy}(f) = \frac{1}{N}\sum_{k=1}^{N} Y(t_{k,} f)\, Y(t_k, f)^* \tag{4}$$

where $X(t_{k,} f)$ and $Y(t_k, f)$ are the discrete Fourier transform of $x(.)$ and $y(.)$ over the time window centered on $t_{k,}$, respectively.

For a chosen frequency f, if we denote $X(t_{k,} f)$ and $Y(t_k, f)$ by vectors $\boldsymbol{u} = [u_k,\ k = 1,\ \ldots, N]$ and $\boldsymbol{v} = [v_k]$ **,** we can see that this is just the formula for Cauchy–Schwartz inequality,[5] which expresses how two vectors are parallel to each other:

$$C(f) = \frac{|\langle \boldsymbol{u},\boldsymbol{v}\rangle|^2}{\langle \boldsymbol{u},\boldsymbol{u}\rangle\langle \boldsymbol{v},\boldsymbol{v}\rangle} = \frac{|\langle \boldsymbol{u},\boldsymbol{v}\rangle|^2}{{\|\boldsymbol{u}\|_2}^2{\|\boldsymbol{v}\|_2}^2} \leq 1 \tag{5}$$

The equality is achieved iff (if and only if) $\boldsymbol{u} = \alpha\boldsymbol{v}$, for some scalar α. The generalization of the Cauchy–Schwartz inequality is the Hölder's inequality,[6] which basically allows for different vector norms:

$$\frac{|\langle \boldsymbol{u},\boldsymbol{v}\rangle|}{\|\boldsymbol{u}\|_p\|\boldsymbol{v}\|_q} \leq 1, \tag{6}$$

$$\text{where} \quad \frac{1}{p}+\frac{1}{q} = 1 \quad \text{and} \quad \|\boldsymbol{u}\|_p = \left(\textstyle\sum_1^n |u_i|^p\right)^{1/p}$$

The equality is achieved iff $\boldsymbol{u}^p = [\ldots |u_i|^p \ldots] = \alpha\boldsymbol{v}^q$, for some scalar α. Now the Hölder's inequality itself can be generalized to parallelism between more than two vectors:

$$\frac{\|\prod_1^m \boldsymbol{u}_l\|_r}{\prod_1^m \|\boldsymbol{u}_l\|_{p_l}} \leq 1, \tag{7}$$

$$\text{where} \quad \textstyle\sum_1^m \frac{1}{p_l} = \frac{1}{r} \quad \text{and} \quad r, p_1, \cdots, p_m \in\,]0, \infty]$$

The equality is achieved iff all $(\boldsymbol{u}_l)^{p_l} = \alpha_k\boldsymbol{w}$, for some vector $\boldsymbol{w}$, that is, they are parallel to the same vector).[7] If we pick the exponents (or equivalently, the norm) to be all equal, we get a more symmetrical expression for parallelism among m vectors:

$$\frac{\|\prod_1^m \boldsymbol{u}_l\|_1}{\prod_1^m \|\boldsymbol{u}_l\|_m} \leq 1, \tag{8}$$

$$\text{where} \quad p_1 = \cdots = p_m = m, \quad \text{and} \quad r = 1$$

Let $\boldsymbol{u}_l = X_l(t_{k,} f)$ be the Fourier coefficient for frequency f for data channel l over the time window k. Then the coherence among all the microphone signals is now

$$C_{xy}(f) = \frac{\sum_{k=1}^{N} \prod_{l=1}^{m} |X_l(t_{k,} f)|}{\prod_{l=1}^{M} \sum_{k=1}^{N} \left(|X_l(t_{k,} f)|^m\right)^{1/m}} \tag{9}$$

To our knowledge, this result has not been used in the open literature. Two examples of its superior performance can be seen in Figs. 6 and 7. Figure 6 shows the data for a hand clap recording in ARL's semi-anechoic chamber. The red curve of subfigure "Detection feature(s)" shows the coherence, averaged over all frequencies, using the standard average over all microphone pairs. The blue curve shows the new coherence using the generalized Hölder's inequality. One can clearly see that the floor noise (or clutter) for the red curve is higher even though both curves achieve the same peak coherence. Thus, the new coherence allows more discrimination against background clutter. Figure 7 shows the data for small incidental signals, possibly coming from moving lab equipment and stepping around the chamber. Again, the clutter is lower for the new coherence and can be better rejected.

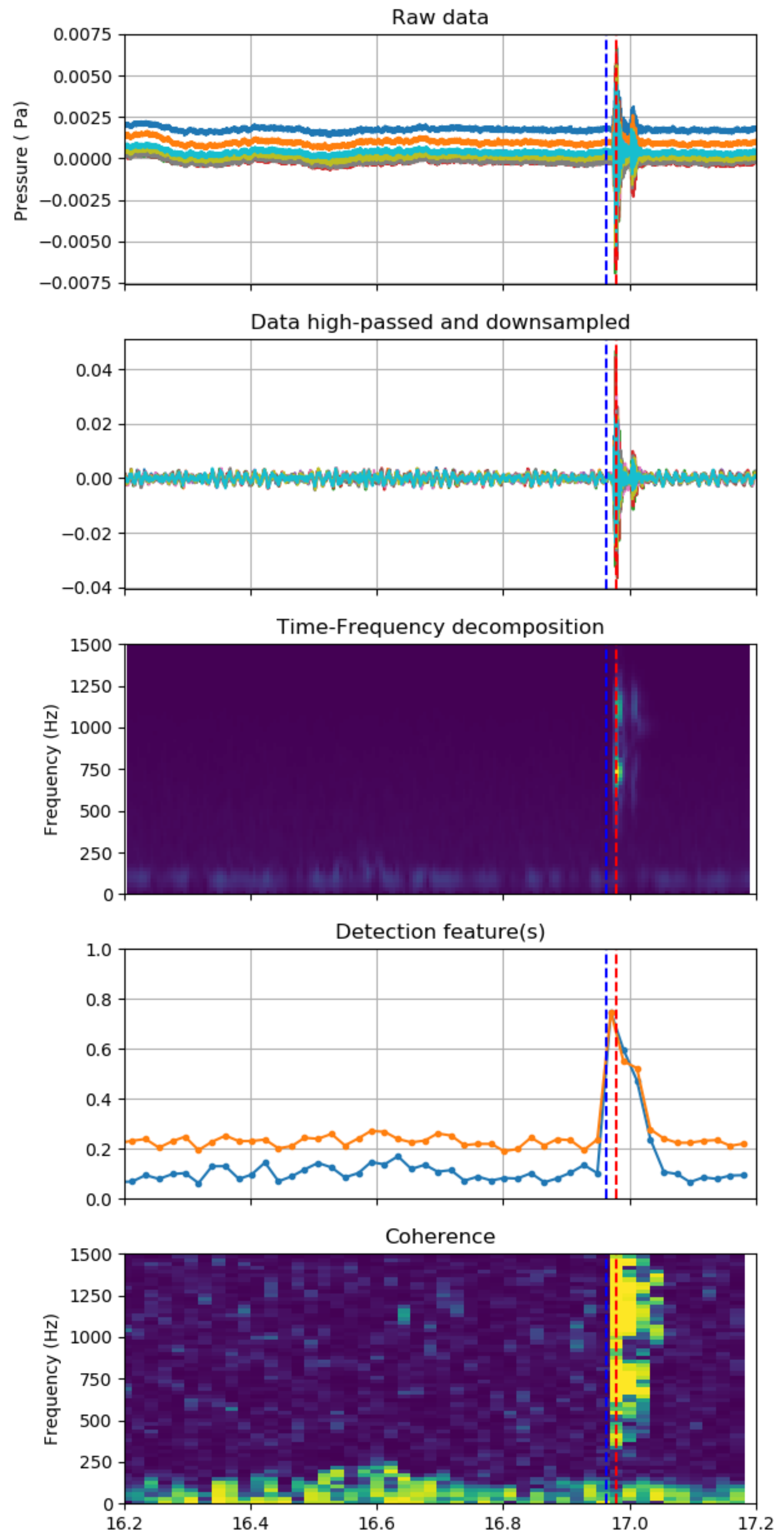

Fig. 6 Coherence for a hand clap

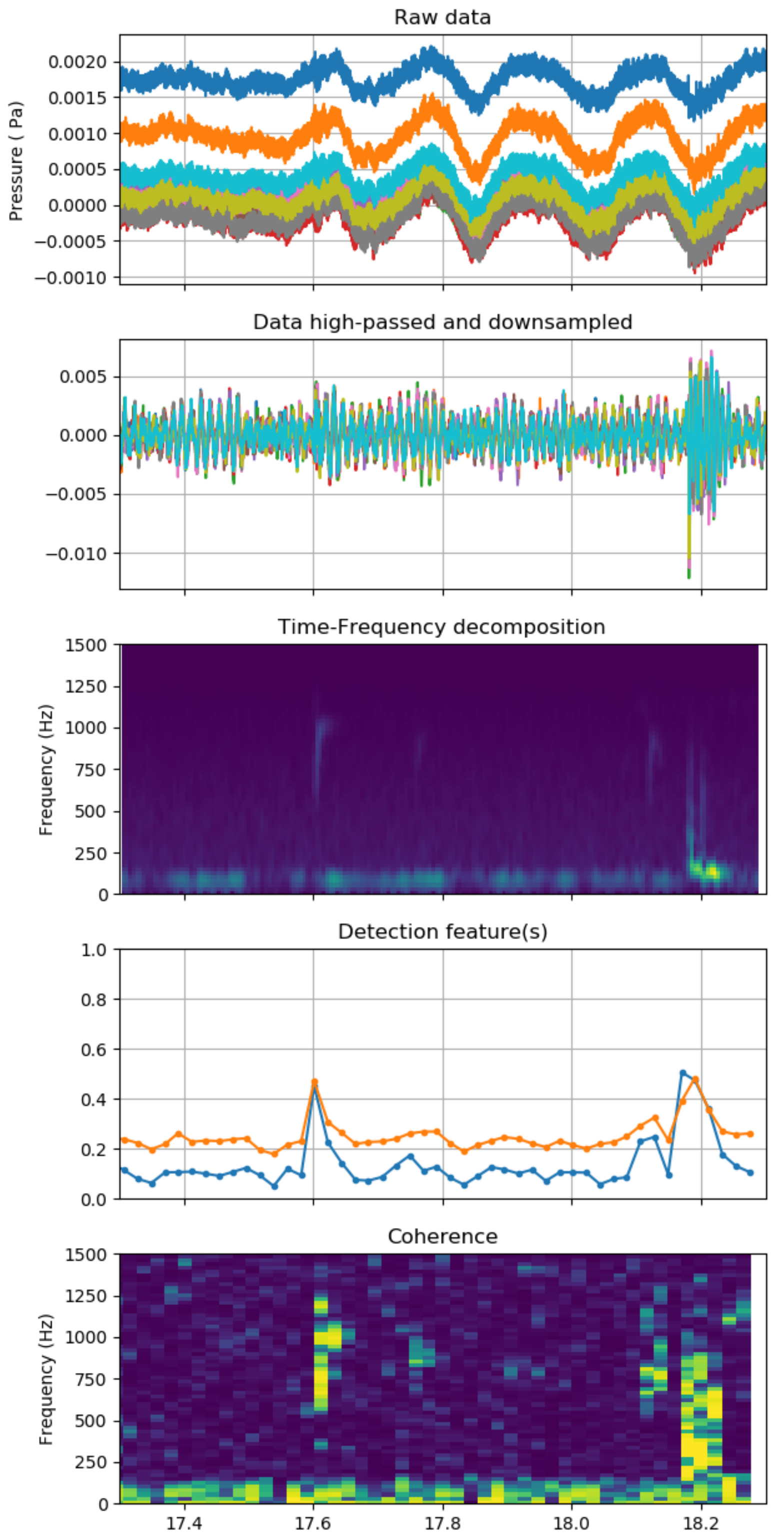

Fig. 7 Coherence for incidental signals

4. Conclusion

From start to finish, the development time took 8 months, including a few field tests. The final product is a portable package of size 10 × 7.5 × 6 inches, weighing 1.71 lb (776 g), and consuming 9 W. The average accuracy is ±3.5° in typical mild atmospheric condition, with a response time around 2 s (delay from the arrival of the acoustic signal to the detection output on the screen). Given the weak 2-core 800-MHz ARM processor, the sampling rate could only be pushed up to 2 kHz while avoiding data packet drops and minimizing response time. The USB interface of the touch screen unfortunately took 15% of the CPU execution time (the processing took 65% to 75%) because of user input polling in Python code.

Performance will likely improve by using numba, the Python just-in-time compiler, instead of Cython. Cython has to call numpy, the Python numerical library, through the Python interface and cannot call numpy directly. This creates a slowdown in repeated calls to numerical routines. Numba plays nicely with numpy by calling it directly, albeit with some restriction on the subroutine options.

5. References

1. Scanlon M, Tran-Luu T-D. Acoustic mortar detection system. MSS Conference; 2003 Sep; Laurel, MD.

2. Tenney S, Mays B, Hillis D, Tran-Luu D, Houser J, Reiff C. Acoustic mortar localization system – result from OIF. Army Science Conference; 2004 Sep; Orlando, FL.

3. Tran-Luu T-D. Technical description of NVAA. Adelphi (MD): CCDC Army Research Laboratory; 2019 Oct. (available from Justin Miller at NVESD or the ARL author).

4. Coherence (signal processing). Wikimedia Foundation; 2020 Apr 2 [accessed 2020 Apr]. https://en.wikipedia.org/wiki/ Coherence_(signal processing).

5. Cauchy–Schwarz inequality. Wikimedia Foundation; 2020 Aug 19 [accessed 2020 Apr]. https://en.wikipedia.org/wiki/ Cauchy%E2%80%93Schwarz_ inequality.

6. Hölder's inequality. [accessed 2020 Apr] Wikimedia Foundation; 2020 Aug 15. https://en.wikipedia.org/wiki/ H%C3%B6lder%27s_inequality.

7. Treibergs A. Inequalities of analysis. Lecture; University of Utah; 2014 Fall. p. 18.

Appendix. User's Manual

A.1 Powering Up

Figure A-1 shows the external components of the Night Vision Acoustic Array (NVAA) unit:

1) Turn on power switch. The screen will turn on the graphical display. However, the CPU is not on yet.

2) Push and hold the start button for 15 s until the LED light on blue. Wait until it starts blinking green.

3) If the LED blinks red, there is no GPS reception.

4) Wait 5 s for the adaptive filter to adapt and start.

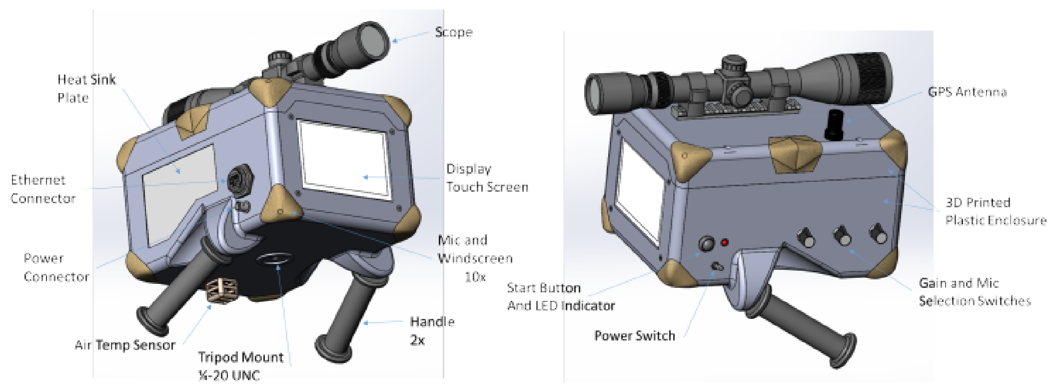

Fig. A-1 NVAA unit with external components

A.2 Graphical Display

The screen (Fig. A-2) shows at all times a set of concentric circles with dots for detected targets. The center of the circles shows the direction where the array is pointing (not absolute North). The innermost circle corresponds to 5° from the main axis of the unit, the next to 10°, 30°, 45°, 60°, and the outermost circle to 90°.

The dots have fix azimuth and elevation angles relative to geographical North, so they will move on the screen if the user changes the pointing direction of the array.

The dots are blue if the target is in front of the array and green if they are behind the array. They are light blue/light green if the signal-to-noise ratio was low. Elevation error is expected to be much higher due to the small vertical aperture and unknown ground reflection during operation.

1) To start dot display, press the Start button.

2) For now, the last three detections are retained.

Fig. A-2 Screenshot of main graphical user interface (GUI) window

A.3 Log Display

The log shows in each column the time of detection (Coordinated Universal Time [UTC]), the azimuth angle in degrees (compass convention), the elevation in degrees, the estimated relative speed of sound at the measured temperature (a value near 1 is better), and the least-square error (LSE; a smaller value is better).

1) To start the Log screen, press Log button.
2) To exit the Log screen, press, Stop button, then Home button.
3) A maximum of 10 lines will be displayed at any time.

A.4 Microphones Configuration

To start selecting microphones (Fig. A-3), press the Configuration button.

Each button corresponds to the numbered microphone. Hit each button to select the corresponding microphone. New microphone selection is not final until one press the Home button.

Cancel the current selection by pressing Reset button.

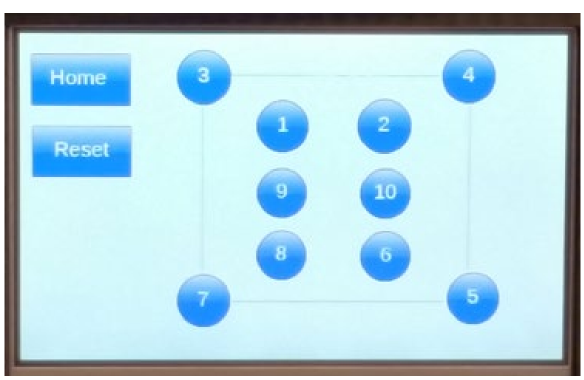

Fig. A-3 Screenshot of microphone configuration window

A.4.1 Caution

1) Since the analog-to-digital convertor (ADC) only has eight channels, while the unit has 10 microphones, one must select which channel (7 or 8) the microphone (6, 8, 9, 10) goes to. Channel selection is done by turning the two knobs on the far side of the unit.

2) Since microphone 6, 8, 9, 10 can go to either channel 7 or 8, during the microphone selection on the screen, the first microphone of the subset (6, 8, 9, 10) selected will get channel 7, and the second microphone of the subset (6, 8, 9, 10) selected will get channel 8.

3) Make sure the software selection (on the screen) and the hardware selection (turning the knobs) match.

A.5 Downloading the Log File to a PC

1) Connect an Ethernet wire from the PC to the COM port of the unit.

2) Turn on the unit.

3) Use FileZilla on the PC to connect to the unit:

   a) Set host as `10.10.0.1`

   b) Set protocol as `SFTP - SSH file Transfer Protocol`

   c) Set Logon Type as `Normal`

   d) Set User as `snickerdoodle`

   e) Type password as `snickerdoodle`

   f) Hit the `Connect` button.

4) On the right pane, navigate to the folder `/home/snickerdoodle/Projects/Results`

5) All log files have name with syntax `nvaa_YYYY-MM-DD_HHmmss.ltx`

6) Click and drag the log file of interest to a folder on the PC (left pane).

A.6 Upgrading the Software

1) Repeat step 1–3 as in Section A.5.

2) On the right pane, navigate to the folder `/home/snickerdoodle/Projects/NVAAv1.0`

3) Click and drag the new files (from the left pane) onto that folder.

A.7 Shutting Down

Before shutting down, press the Exit button. This will ensure the file log has time to close.

List of Symbols, Abbreviations, and Acronyms

3-D	three-dimensional
ARL	Army Research Laboratory
CCDC	US Army Combat Capabilities Development Command
CONOPS	concept of operation
CPU	central processing unit
DC	direct current
DoA	direction of arrival
FFT	fast Fourier transform
FO	forward observer
FOV	field of view
FY	fiscal year
GPS	global positioning system
IMU	inertial measurement unit
JETS	Joint Effects Targeting System
LED	light-emitting diode
LOB	line of bearing
LSE	least-square error
MASINT	Measurement and Signature Intelligence
MDUSA	Multi-Domain User Sensor Architecture
NVAA	Night Vision Acoustic Array
NVESD	Night Vision and Electronic Sensors Directorate
UTAMS	Unattended Transient Acoustic MASINT System
UTC	Coordinated Universal Time

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
1 (PDF)	CCDC ARL FCDD RLD DCI TECH LIB
9 (PDF)	CCDC ARL FCDD RLS SA M SCANLON L SIM H VU D GONSKI J GOLDMAN W ALBERTS FCDD RLC CA TD TRAN-LUU E MARK L KAPLAN