SAM Correlation Analysis Documentation

Steven Doran and Navaneeth Poonthottathil

August 2022

1 Introduction

This document provides a description and explanation of the experimental techniques, scanning procedures, and coding infrastructure used in the Scanning Acoustic Microscopy (SAM) correlation analysis of the 100 P3 Deep Underground Neutrino Experiment Liquid Argon ASICs (DUNE LArASICs) Thermal Stress Test Study conducted at Iowa State University (ISU). The motivation for using SAM to non-destructively evaluate the LArASICs is to investigate internal changes induced by thermal-related stress. SAM is widely used as a non-destructive technique in industry. The tools, techniques, and procedures employed in the stress test study can be applied to other DUNE ASICs and other electronics. The structure of this document is as follows; section (2) provides a general description of scanning acoustic microscopy and its applicability in correlation analyses, (3) details the experimental procedures for taking acoustic data, while (4) and (5) provides a description of the correlation analysis used in both MATLAB and Python to identify and quantify changes observed within the ASICs due to cold cycling.

2 Scanning Acoustic Microscopy

A scanning acoustic microscope enables non-destructive testing and evaluation of the internal circuitry, producing acoustic waveform profiles for a grid of spatial points on the target. A schematic of scanning microscopy is given in Figure 1; a piezoelectric transducer, attached to a buffer rod, converts an electrical voltage impulse into an ultrasonic pulse. The ultrasonic waves travel down the buffer rod (or delay line), with a focusing lens, and they are coupled using a fluid into the object being examined. Pulses are then reflected back into the buffer rod from interfaces and other anomalies. The ultrasonic signals enable the internal structure of the object under test to be mapped in 3-D. The SAM, used in this study, uses water as the couplant. It has a nominal transducer frequency of 10 MHz and this provides a 130μm resolution. When the transducer is scanned (2-D) a raster scan is collected across the surface of the entire ASIC, corresponding to over 10,000 waveforms. A single pulse-echo voltage waveform has "echo's" that are due to acoustic impedance changes throughout the depth of the ASIC; depth and time are both characterized by the same coordinate. SAM analysis thus yields a three dimensional reconstruction of the ASIC's acoustic properties, which are synonymous with the physical structure of the ASIC's internals.

By comparing the waveforms at each spatial point between two different measurements ([0] times cryocycled vs [5] times cryocycled for example), the correlation can be calculated for a specific window of time (specific depth range) and a colormap of correlation can be created, highlighting regions of anti-correlation between the measurements. These regions of anti-correlation correspond to differences in the acoustic properties of the materials; since the measurements are of the same ASIC, the change displayed through correlation mapping is representative of changing internal structure due to the effects of cold cycling. SAM analysis therefore makes it possible to monitor the mechanical integrity of the ASIC or any electronic throughout thermal-stress testing. An example of a SAM correlation colormap (referred to as a correlation plot from here on) and an example of a single data point (RF-data record), also called an A-scan, for a region of anti-correlation is shown in Figure 7.

The SAM data can be collected at different time points and correlation analysis can be performed for

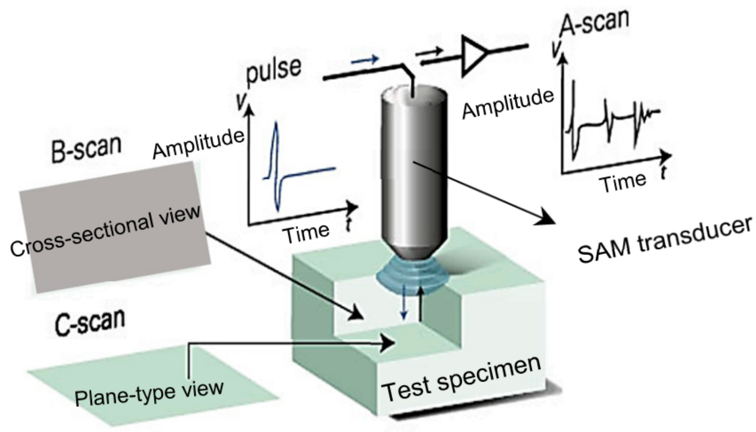

Figure 1: Schematic of a Scanning Acoustic Microscope.

each ASIC looking at data taken after several different cryocycle milestones. Correlation plots between [0] cold cycles and [100] cold cycles show a response that correlates to physical changes in the ASICs after being exposed to 100 cold cycles. The number and size of the anti-correlation regions (ACRs) are representative of the physical changes induced from thermal/cryogenic exposure. This methodology can be easily extended to other DUNE ASICs, as they will also be submerged in liquid argon for years. The SAM correlation analysis detailed in this document is perfectly suited and can be readily used for assessing the quality and integrity of the plastic packaging which encases the DUNE ASICs. In addition, other electronics from other disciplines could benefit from a similar analysis using potentially different stress tests. Overall, the SAM correlation analysis developed for the DUNE LArASICs has widespread applicability. A general overview of the methodology of the thermal stress test study and how the correlation analysis is performed is shown in Figure 2. As can be seen from the figure, acoustic data between two different measurements, before and after cold cycling, yield correlations across a grid of spatial points. The differences between the measurements at various grid points shine through as regions of anti-correlation.

3 Experimental Procedures for Recording Acoustic Data

The SAM apparatus used is a 3-axis scanner capable of acoustic imaging many targets (ASICs) at once, located at the Applied Science Complex (ASC) at Iowa State University. As discussed above in Section (2), the particular setup relies on an half-inch immersion probe transducer, centered at 10 MHz. The setup for the targets was created using the case that the ASICs were transported in from Brookehaven National Laboratory. The plastic fixture is designed to hold the ASICs in place with minimal freedom of movement. The plastic fixture was glued and placed at the bottom of a plastic water tank, and wedged in place by a metal plate. The fixture had to be cut to fit horizontally within the water tank, but is capable of holding as many as 78 LArASICs. For this work, 30 LArASICs were scanned at a time (aside from the last set of 10). A picture of the plastic fixture can be found in Figure 8. The SAM was set up to scan in the x and y directions at a fixed z (height).

In preparation of scanning the samples, calibrations must be made to ensure the fixture is properly aligned with the scanner and that the scanner is at the appropriate height. In addition, settings like the gain and scan speed must be fine-tuned to yield as strong and consistent of an acoustic signal as possible. Therefore, there are two primary elements of calibration; firstly, optimizing the scan settings on the computer and the external pulser/receiver. Secondly, identifying the scanning origin and the area of the scan, as well as ensuring the pitch and alignment of the fixture is ideal. A screenshot of the software, the external pulser/receiver, and the overall scanner and physical setup are shown in Figures 5, 3, and 4, respectively.

Before the procedure is outlined, a **huge** thanks to Dan Barnard of the Iowa State Center for Nondestructive

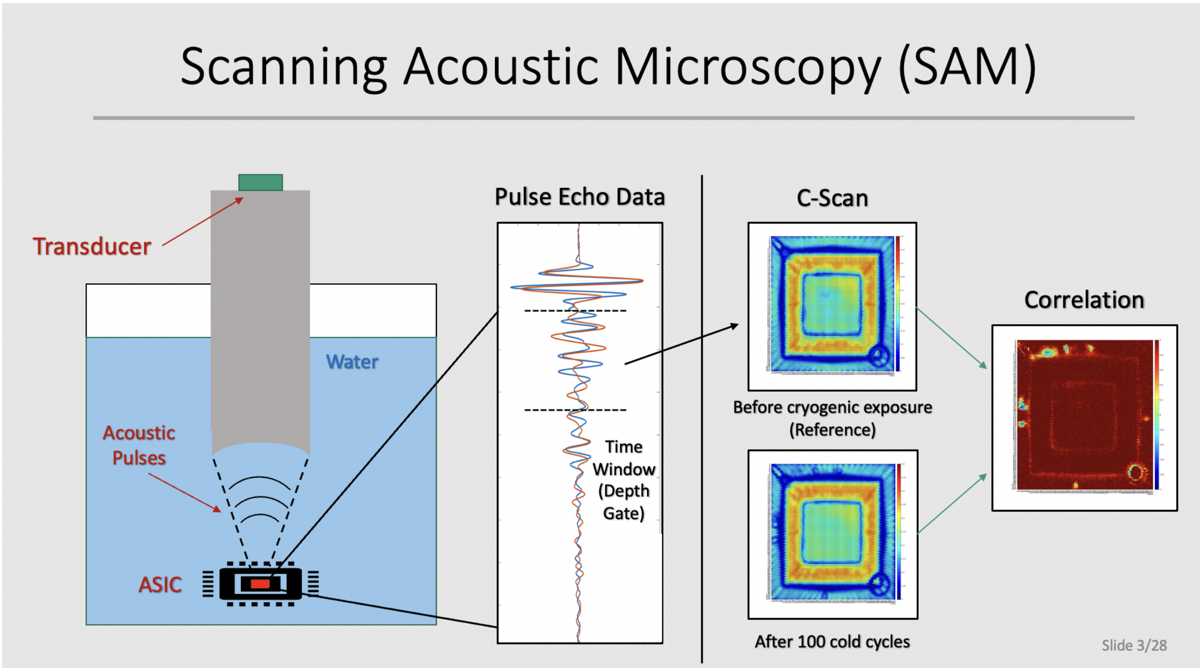

Figure 2: A diagram detailing the correlation analysis developed using SAM.

Evaluation for his expertise regarding scanning acoustic microscopy. His help was instrumental in taking data and standardizing the procedures for this work.

Alignment and Setup procedure:

The tank should cleaned prior to filling, and be filled with distilled water. Water in the tank should be left to set at least overnight to eliminate most of dissolved air/gas.

For the initial pulser/receiver settings, this may vary from scan to scan. For this work, these were the optimized parameters used:

- PRF = 2.00 KHz
- Energy = 12.5 μJ
- Damping = 50 Ohm
- HP filter = 1 MHz
- LP filter = 35 MHz
- Input attenuation = 0.0 dB
- Output attenuation = 11.0 dB
- Gain = 20.0 dB
- Sensitivity = 54.9 dB

These parameters should be optimized to ensure the acoustic signal is as strong and clean as possible. The parameters will change with different pulser/receivers as well as different transducers. The pulser/receiver used in this work is the Computer Controlled Pulser/Receiver, Model 5800 from Panametrics.

Figure 3: The pulser/receiver used in this work.

The following descriptions provided about calibration and alignment using the computer at the ASC are based on that computer's software. This will clearly change if a different computer software is used. The overall themes of calibration are transferable in other extensions of this technique, but the individual values of the parameters must be discovered through optimization. Once an appropriate set of parameter values are found and used, their values should be documented so that repeat measurements will share the same settings. The parameters quoted for the software used in this work follow a screenshot of the software display shown in Figure 5. Each parameter value stated corresponds to the inputs seen on the display. See the figure for where exactly to input the values. Also, note the values on the figure are not the ones used for this work; those specific values will be stated in the text.

The focal length of the delay line and transducer in this case is two inches. The height (z axis) must be adjusted accordingly. The largest variable in the z-axis value will be the tank and the stand holding up the tank. For this work, a small plastic table/stand was placed underneath the water tank. The associated z-axis value was set to 65.5. In order to find the correct scanning height, the signal amplitude off the target should be maximized.

The scanners speed decreases when it changes directions. The scanning axes were selected such that the longer axis (6 ASICs) would be scanned across (y direction), while the other axis (5 ASICs) would be the direction worked towards (x direction). This strategy should be adopted to save time, as well as eliminate the effects of backlash from the motor.

With the tank and fixture oriented as shown in Figure 4, scan along the left and top continuous plastic edge. Bump tank as needed to align edge with y-axis and re-scan – repeat until aligned with axis. This work often used 3 ASICs along the x and y axis edge to ensure the fixture was rotationally aligned with the scanning axes. **This is critical**, as any rotational differences between measurements cannot be fixed in the code infrastructure due to dimensionality of the filetypes. Also, ensure that the ASICs signal amplitudes are not decreasing as the scanner moves along either axis. Any signal decrease corresponds to the samples moving out of the focal length window and hence change in the pitch of the fixture (a different height between ASICs). It is advised an ASIC be placed in each corner to calibrate the overall height change across the fixture. Ensure the effect is minimized, as this change will induce artificial anti-correlations in the data due to phase shifts in the waveforms (from being at different heights) that cannot be fixed in the analysis. This is why it is also important to have a repeatable setup and to document the z axis height for each measurement. In this work, no significant difference in pitch was observed, so 'ballparking' a z-axis value based on the amplitude of the signal and then maintaining that value between measurements is adequate.

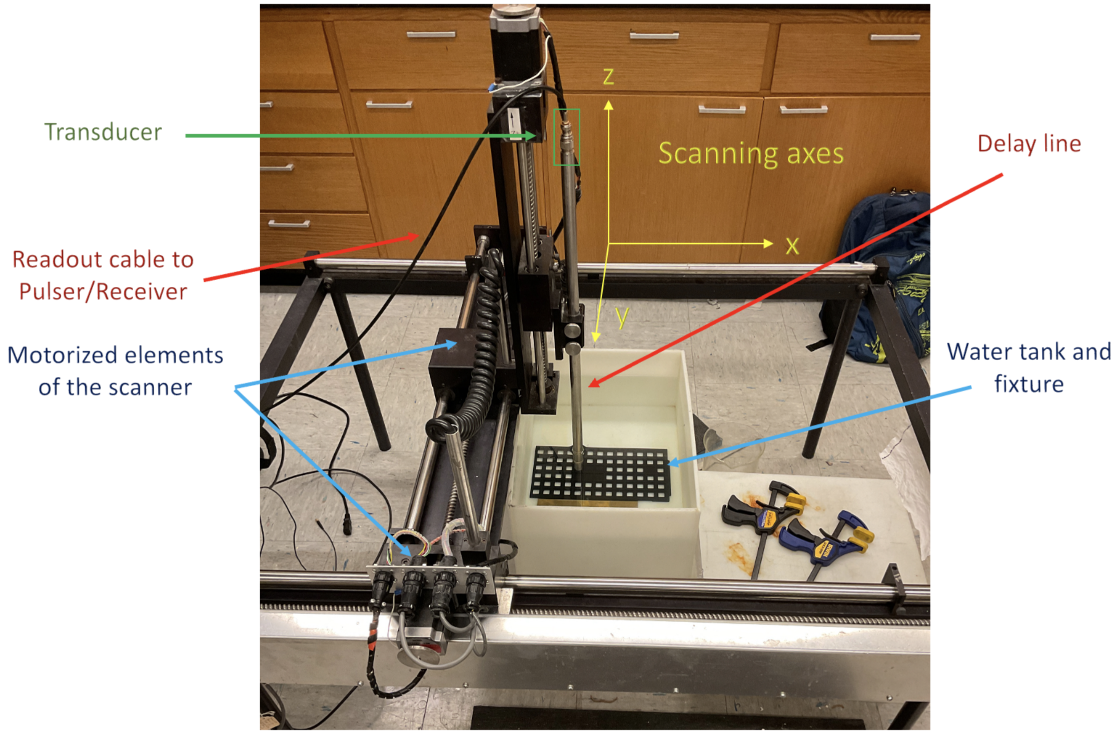

Figure 4: The setup of the scanner used at the ASC.

Scan the upper left pocket and zoom in to time-of-flight scan to establish horizontal and vertical cursors at top and left edges of pocket, as in Figure 6. Use the Track command to move to the cursor crossing and Reset X and Y axis to read zero. This is now our reference corner (0,0). The selection of the reference corner (down to the same pixel) is not always the same between measurements, as the ASICs position can be slightly different due to the freedom within the fixture. In general, ensure the same general area of the lower part of the ASIC is used. Small translational differences in scans can be fixed in the code infrastructure.

Move probe on X and Y axis to the center of the fixture grid location, as also seen in Figure 6. Note the cursor positions by hovering over the cursor lines. The selection of the center may vary from scan to scan - this is okay. Extra edge room will be scanned to account for differences in measurements' origin and centerpoints. Once selected, this is now the center of the scan area. For this work, the center position is (1.859", 2.317") in (x, y). The center position is selected for the 6 ASIC by 5 ASIC fixture scan (see Figure 8), but can be adjusted if more ASICs are planned to be scanned. Move to the center before starting a scan.

Since the dimensions of the fixture are known, the scan lengths can be set in order to scan all ASICs in the fixture. For this work, scanning lengths of 3.9" (scanning in x direction) and 4.8" (indexing in y direction) were used. The total scan area should encompass the entire sample (all ASICs). It is advised the scan should be extended by a half inch on each side of the fixture. This was not utilized in this work, but is advisable since the code must align each ASIC data between measurements. This requires extra room around the fixture in order to ensure each ASIC is translationally aligned between measurements (see Section (4)).

The scan speed and index speed used was 0.25. Scanning 30 LArASICs takes around 4 hours and 30 minutes. The scanning velocities may be increased to decrease total scan time, but be careful about missing data. The values of the scan speed were optimized for this work. Missing data will come in the form of 'NaN' pixels on the computer scanning readout. Make sure to reduce the data missed as much as possible. Note that some pixels will ultimately be lost in the data for whatever reason (and randomly), but should be a negligible percentage of the total pixels available.

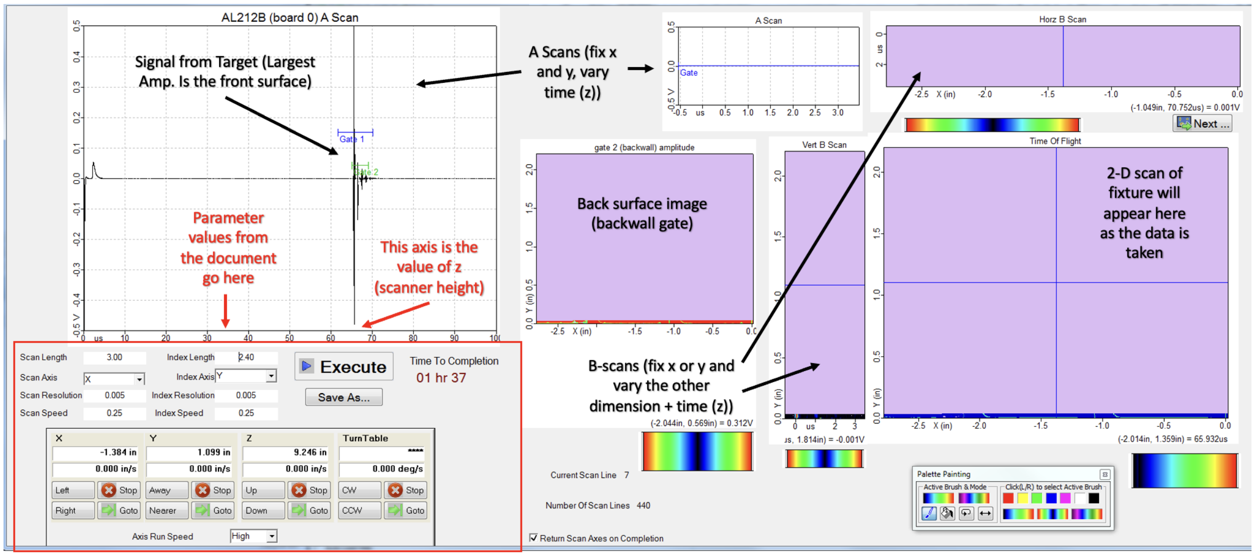

Figure 5: Typical display from the acoustic data-taking software on the computer used at the ASC.

Other relevant settings on the computer include the the sampling rate, which was set to 250 in the internal settings, as well as the follower gate threshold set to 0.1. The gate length (for the amplitude signal) was 2.644 microseconds, corresponding to a range of [-0.196 to 2.448] microseconds. Physically, this is gating just in front of the initial front wall amplitude (largest amplitude seen in the A-scan on the computer) to a region where there no longer is any noticable signal (past the reverberations). This can be seen in Figure 5 in the top left A-scan.

It is advised that a first row be scanned to ensure the settings and calibration chosen is adequete. If the scan looks good, then proceed to reset the scanner and take data. Following the completion of the scan, the machine will stop and return to the intial scanning location. The computer will display the associated 2-D surface scan (C-scan), as well as the B-scans (both horizontal and vertical). The acoustic data taken can be saved as a .sdt filetype and exported to a external drive. Make sure to bring an external drive, as the primary computer (maybe) doesn't have internet access. The total size of the file (30 LArASICs) will be around 1 GB.

Acoustic scans were done in the morning, following calibration, and the scan was subsequently completed sometime in the afternoon. After the scan completes, verify the scan is completed and looks good, then remove the ASICs from the water. Be careful and ensure you are grounded prior to handling the ASICs (or use rubber gloves). ESD has been known to induce functional failures in the chips. Air drying the ASICs within a transportation plastic case is fine; make sure to transport the ASICs in an anti-ESD bag. This goes for all electronics being scanned with SAM. The same level of precaution should be maintained for placing the next round of ASICs in the fixture. Ensure the water tank is not moved while replacing the ASICs, as the scanner will have to be re-calibrated. It is recommended the ASIC positions are documented prior to scanning. Also ensure the ASICs are snugly positioned within the fixture, pushing each of them to a consistent corner (if the fixture have some freedom of movement). It is wise to recheck each scanning direction's alignment prior to performing a scan.

A few important things to note regarding the setup and the subsequent correlation analysis are the following; the entire setup should be stable and secure enough to not move during the SAM scan. Any movement can lead to faulty data collection and thus a faulty correlation analysis. The tank and fixture must be positioned as close to as exactly rotationally aligned as possible with the axes of the scanner. As will be discussed, any rotational alignment errors are difficult to correct for in the correlation analysis. The overall placement of the LArASICs should be, but don't necessarily have to be, in the same position between

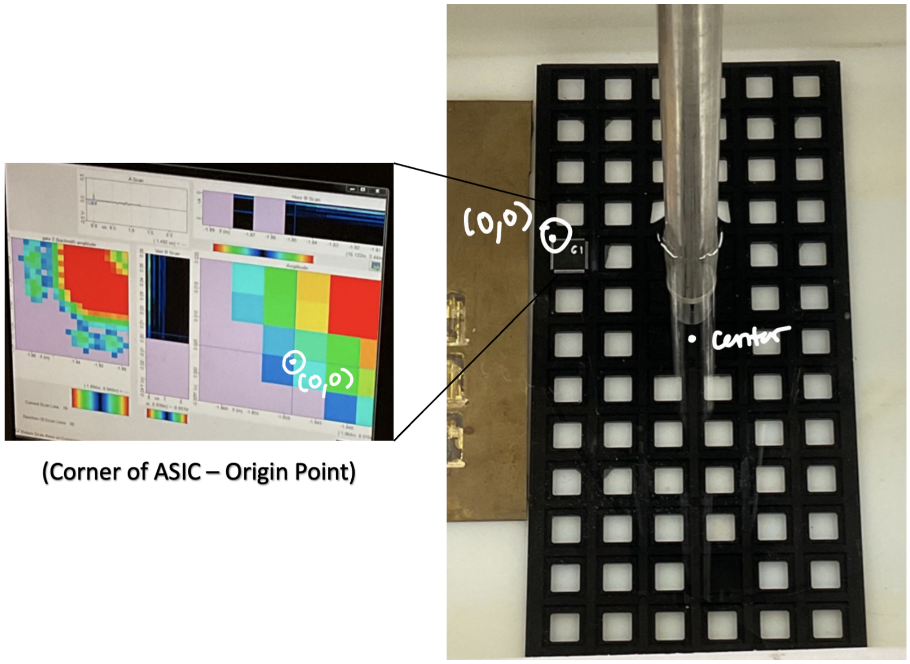

Figure 6: A diagram showing the selection of an origin point (0,0) for an acoustic scan.

measurements. Obviously, to reduce systematics the scan should be conducted with the ASICs in the same position as the previous scan. However, the code has the ability, as will be discussed, to fix translational differences between datasets. The fixture must be precisely flat. Any incline will lead to artificial phase differences in the waveforms and thus artificially-induced anti-correlations. In extending this technique to other electronics, note that the overall takeway is that there must be a repeatable, stable fixture of sorts to place the sample targets in. This fixture must be flat, aligned with the axes, and have a repeating structural pattern. This way, it is easy to automate the correlation analysis in the code infrastructure. It is wise to label the front surfaces of the targets, so that their positions in the initial measurements can be documented. Since the regions of interest for this study are below the surface of the ASICs, any differences from writing their numbers on the surface will not affect the correlation analysis. Obviously if the area of interest in the electronics is topological, then a different identification scheme should be reconsidered.

Something that was noticed in developing the experimental techniques of the scanner for the DUNE ASICs was the issue of corrosion/oxidation of the metal external pins during scanning. The initial fixture developed for this project was milled and made of brass. Upon submerging and scanning the LArASICs in water for hours, it was discovered the external pins began to react with the brass fixture. This produced a layer of corrosion/oxidation on the external pins which affected the functional performance of the chips. The oxidation/corrosion was later removed, but it is important to investigate how the material of the fixture could potentially react to other metals on the target when submerged in water for extended periods of time. A plastic fixture eliminated any corrosion/oxidation issues for the DUNE ASICs, and is recommended for any extensions of this technique.

Lastly, it is important to keep the samples clean from oil, dirt, or grease. Also, it is important to keep the tank clean prior to scanning. Any dirt-related impurities could lead to missing areas of data during the scans. Prior to gluing the fixture on the bottom of the water tank, the bottom of the tank was sanded and cleaned to eliminate contaminants and ensure the bottom was as flat as possible.

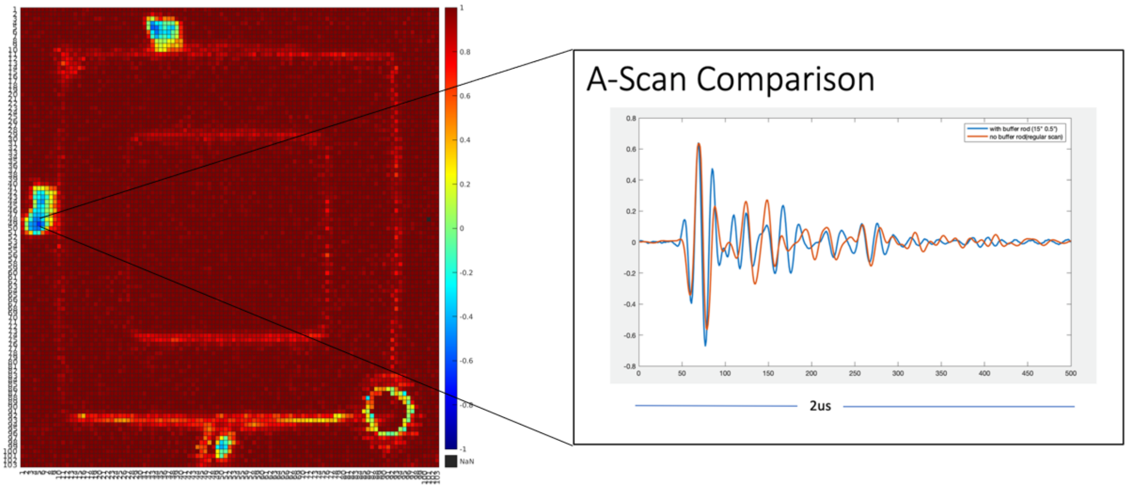

Figure 7: A correlation colormap highlighting areas where the acoustic data is different between measurements, with an associated A-scan of an individual point in x and y.

4 MATLAB - Correlation Analysis

The acoustic data recorded is stored as a .sdt file. For a 30 LArASIC scan, the file size is roughly 1 GB. MATLAB is well suited to handle large files, but be cautious in extending this analysis to a different coding language (in total, the code may be required to load in up to 16 GB or more of data). Though efforts are underway to shift this data type and analysis from Matlab to ROOT, the following description is given for the Matlab analysis used in the Thermal Stress Test study of the 100 P3 LArASICs. The .sdt filetype contains acoustic data in three dimensions; two spatial dimensions detail the length and width of the ASIC, with one time dimension specifying the acoustic waveform received at each spatial grid point. Time is synonymous with depth for this analysis, as the time of flight of the signals yields the acoustic information through the depth of the ASIC. A full waveform through the depth of the ASIC exists for each spatial grid point. The spatial resolution for this analysis is around 130 micrometers, corresponding to nearly 10,000 gridpoints per LArASIC. This resolution allows for the localization and identification of anti-correlation regions when comparing data before and after cold cycling. Each gridpoint waveform is known as an A-scan; fixing two spatial dimensions while varying a third yields an A-scan. Figure 7 shows a diagram of the correlation colormap and the associated A-scan of an individual pixel point. By comparing the waveforms at each pixel point, a correlation analysis can be performed.

The purpose of the MATLAB code is to perform a correlation analysis (shown as a colormap in Figure 7, left) that highlight regions where the acoustic waveforms are not correlated between measurements. Regions of low-correlation are believed to be places where internal changes have been induced from thermal stress. The anti-correlation between two separate measurements, before and after cold cycling, represents changes in the acoustic properties of the material. The MATLAB analysis that is detailed below produces these correlation colormaps for each ASIC, across all cold cycle milestones. The entire process is largely automated, but will be discussed in detail below. Extension of this code to other DUNE ASICs or other electronics in general is straightforward, as the tools utilized are easy to use and have widespread application for non-destructive evaluation.

Below, a full description of the analysis and code for identifying areas of anti-correlation in electronic chips is provided in two parts. The first part (this section) produces the correlation colormaps (referred to as plots from here on), which are useful is identifying changes within the electronics. The second part (next section) takes the correlation plots and runs them through another code infrastructure, OpenCV, using Python to quantify the regions of anti-correlation. Further analysis can then be performed in more detail.

The MATLAB code **Full_Slicer_Finalized.m** is discussed here. Though the code has comments embedded, a more detailed description is presented. The general structure of the code is as follows:

1. The code reads in the .sdt files produced from the acoustic scans. For this specific iteration of the code, all data files from all ASICs across four cold cycle measurements are read in. For each cryocycle and ASIC dataset, the .sdt files are stored in arrays which are sliced on an individual ASIC basis. The code will loop through each ASIC until complete.

2. Next, it is important to only include regions of interest in the analysis. External pins are not of interest and must be excluded. The code identifies the edges of the ASIC, separating the internal area of the ASIC from the pin region.

3. To minimize anti-correlations from translational differences in the data, the code then overlays the left and top edge between cold cycle data arrays, performing an alignment procedure. This ensures that the correlation analysis runs with two data sets which are precisely overlaid (to a 1 pixel (130 micrometer) level resolution).

4. A correlation analysis is conducted at each grid point between two sets of measurements. The correlation analysis gates a certain time window (depth range), yielding a single value at each spatial point for the correlation between waveforms.

5. The correlation values are stored and displayed as a correlation colormap. A negative threshold correlation plot (detailed later) is also produced, which is exported and used in the second part of the analysis (in Python).

6. This analysis is run for all ASICs between each set of cold cycle measurements. It also can perform the correlation analysis between different time windows (depth regions within the ASIC). This is used to produce the correlation matrix plots.

The first step of the code is to read in the .sdt files. The .sdt files for this analysis contain a grid of LArASICs which were acoustically scanned. Shown in Figure 8 is an example. The real fixture is shown next to an amplitude plot of the data. As will be detailed, the regions between ASICs needs to be properly removed in order to yield sensible correlation results. In addition, each ASIC's position varies slightly (by a few pixels) due to some freedom of movement within the plastic fixture. A careful alignment between datasets must be performed.

Each .sdt file contains a set number of LArASICs. For this analysis, 30 ASICs are scanned (maximum) for each .sdt file. One .sdt file contains only 10 ASICs (since 100 in total were scanned). Therefore, the code must loop through four different .sdt files per cold cycle. Since correlation analysis is done between measurements, a total of four different cold cycle measurements must also be loaded for each ASIC ([0], [5], [20], and [100] cold cycles). In total, there are 16 .sdt files loaded in the code (hope you have enough RAM!). The second part of the correlation analysis (below) utilizes the time-of-flight window 100-150 microseconds. This is the depth range where most of the regions of anti-correlation are observed. To produce different depth ranges, the code can be edited to loop through different time windows, as can be seen in the commented-out array. To produce the correlation matrix plots, the code must loop through various time windows. If fully utilized, the code will start with one ASIC in a given .sdt file (say ASIC 20 of the .sdt file ASICs 20-49 cold cycle [0]). It will then go through each ASIC (30 in total) individually, performing a correlation analysis by first looping through the time-of-flight windows of [1:100], [100:150], [150:200], then finally [200:300] microseconds. This is done initially between [0] cold cycles and [5] cold cycles, but is then repeated to do [5,20] and [20,100]. A separate, similar code is used to only do the correlation analysis between [0,100], but this code can be easily changed to accomplish this.

In summary, the loop is as follows:

1. Loop through each ASIC in a given dataset (either 30 or 10).

2. Loop through each time gate.

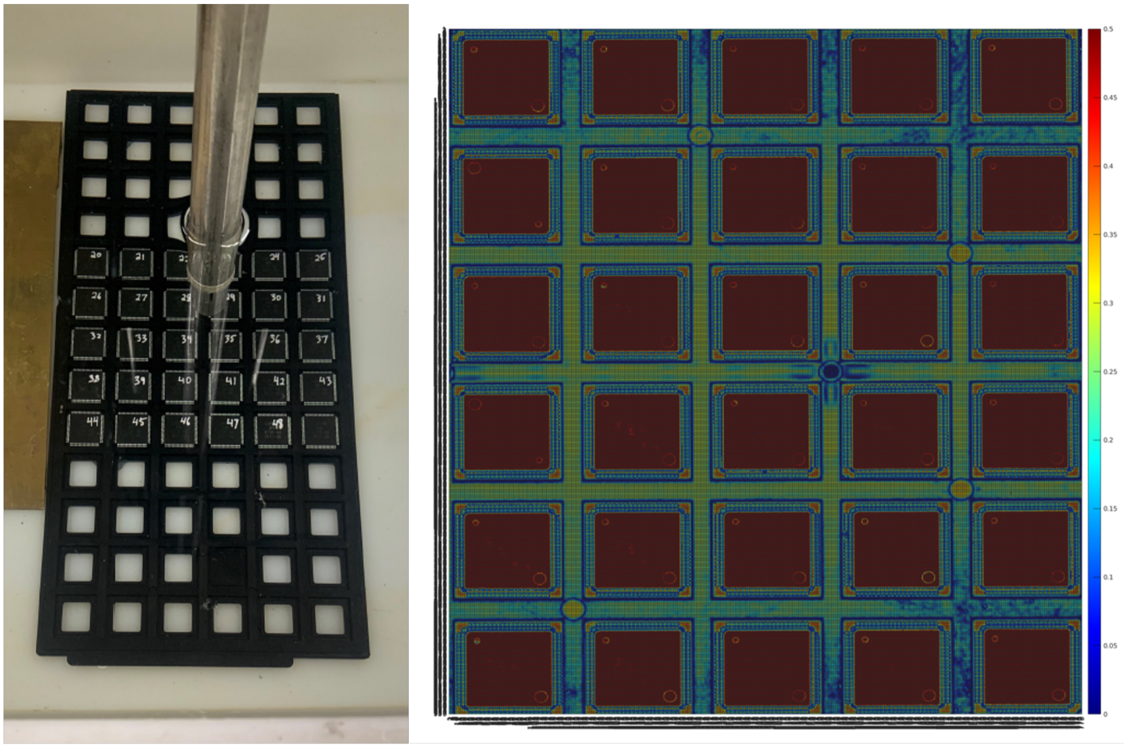

Figure 8: The fixture used for taking acoustic data with the ASICs placed. The corresponding acoustic scan is shown to the right. Note the x and y axis are flipped for the acoustic scan relative to the picture on the left.

3. Loop through each cold cycle measurement ([0,5], [5,20], [20,100]).
4. Loop through each set of ASIC files (20-49, 50-79, etc. . .).

The first step of the loops is to first load in the data for the front wall of the ASIC (TOF window [1:100]). This corresponds to the largest amplitude of the waveform data, which also is the strongest reflection recorded. This front surface provides a great outline of the ASIC, which as we'll see, allows for the data to be constrained to only the relevant internal region while ignoring the external pin regions and the gap between ASICs. The fixture used for holding the ASICs during scans and the associated amplitude plot can be seen in Figure 8. After loading in the front-wall time window data, the code must next properly 'find' an ASIC. This is easy for the first ASIC in a fixture but grows in complication for the final ASICs. This is due to three things; firstly, the ASICs have some freedom of movement within the fixture. Though this movement is very small to the human eye, it results in the ASICs being shifted by a few pixels. In addition, the fixture may be slightly rotated, with respect to the scanning axes. The bottom of the sixth ASIC in a row could be as many as five pixels below the first ASIC, due to an improper rotational alignment. For an individual ASIC correlation plot, this effect typically is less than one pixel so it does not significantly affect the correlation analysis. However, this effect accumulates as you move down the fixture. These two effects must be accounted for when 'jumping' from one ASIC slot in the fixture to another. These two effects are dwarfed by the third problem in looping from one fixture slot to another, across the entire fixture: data alignment. Due to the large time intervals between SAM scans in this study (due to functional tests in RT and LN2), the ASIC fixture used in SAM imaging at the Applied Science Complex was moved after each scan. The scanning alignment thus had to be re-calibrated before each measurement. For each scan, a common origin point is specified on the ASIC fixture, but this origin point varies for each scan (ASIC is not necessarily in the exact same physical spot down to a single pixel). In addition, physical movement of the fixture, other human-induced errors, and backlash from the SAM scanner leads to each .sdt file type spanning slightly different areas. This area is less than a centimeter on each side, but in the .sdt files this

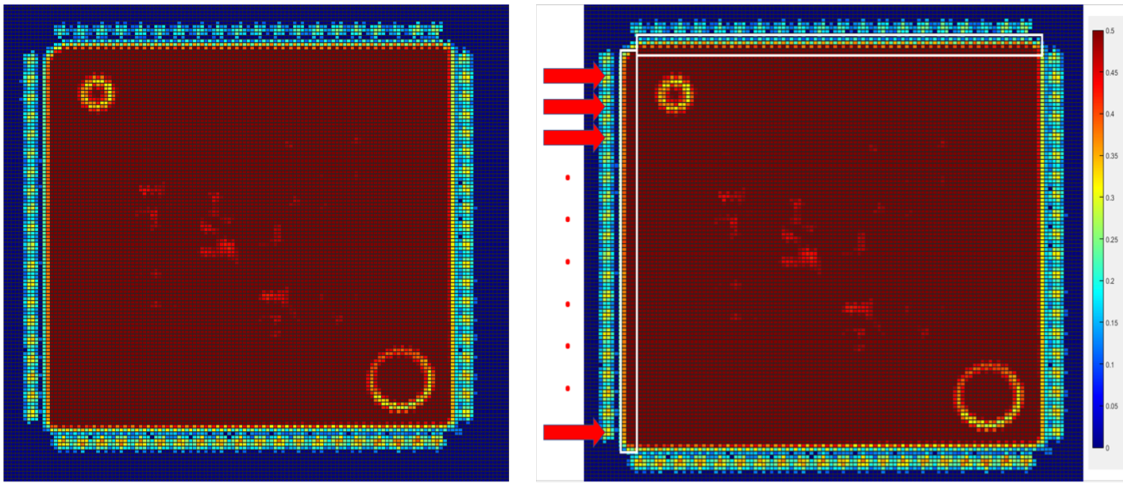

Figure 9: The alignment proceedure used in the MATLAB SAM correlation analysis to ensure the ASICs are positioned correctly between measurement data. The algorithm first walks across the x axis for a given y axis value, looking for a value of the acoustic amplitude large enough, denoting the position of the front surface. This process is repeated down the edge and in the other direction (y) to find the average edge positions. This is used to properly align the ASICs between measurements

corresponds to as many as 10-15 pixels. Since the algorithm loops through each fixture slot, starting from some zero value (first ASIC in the fixture), this loop may correctly capture an ASIC for a specific cold cycle measurement, but will 'grab' part of the neighboring plastic fixture on another cold cycle measurement due to the difference in alignments. The ASICs must be exactly positioned relative to each other in order to yield a sensible correlation analysis. The difficulty arises in conducting this overlay automatically for each ASIC in a fixture. An alignment procedure, detailed below, is utilized to overcome this difference in alignments. However, since the datasets require a shift in the x or y direction by as many as 10-15 pixels, a 10-15 pixel "cushion" is required around each ASIC. In addition, the plastic fixture containing the ASICs can be falsely tagged as an incorrect surface during the alignment procedure. Therefore, some manual adjustment of the code is required to correctly 'capture' each of the 30 (or 10) ASICs, while excluding any of the plastic fixture.

The code is optimized to loop through the specific plastic fixture used in this analysis, namely a 5 row, 6 column fixture but can be extended to any holding fixture for virtually any electronic chip. The 'parameterization input 2' section of the code is carefully calibrated to step through the ASIC fixture for each dataset, plucking the ASIC from the rest of the plastic fixture. Some fine tuning was required to perfect each step, and some further fine tuning is required to extend this analysis to other electronic chips with different physical setups.

Next, the data is concatenated with the MATLAB 'squeeze' command in order to properly dimensionalize the array for analysis. During concatenation, the absolute maximum of each spatial grid point (the maximum amplitude value from a given waveform) is used (absolute since the waveform contains inversions). The values run from 0 to 0.5.

As discussed above, an alignment procedure can be utilized to perform a proper correlation analysis of two distinct datasets. The identification of the outlines of the ASIC is a very important step for this alignment procedure, and for the correlation analysis as a whole. It is required in order to yield non-fallacious anti-correlation regions between measurements. The next step of the code, and the overall process of identifying the outline of the ASIC is shown in Figure 9. As can be seen, the left/right and top/bottom edges of the amplitude plots show a sharp transition between the pins of the ASIC and the front surface. The transition is sharp enough to use as a proxy for the outline of the ASIC. Upon loading a given area of data (discussed above) which contains an ASIC and no neighboring plastic fixture, the algorithm first searches

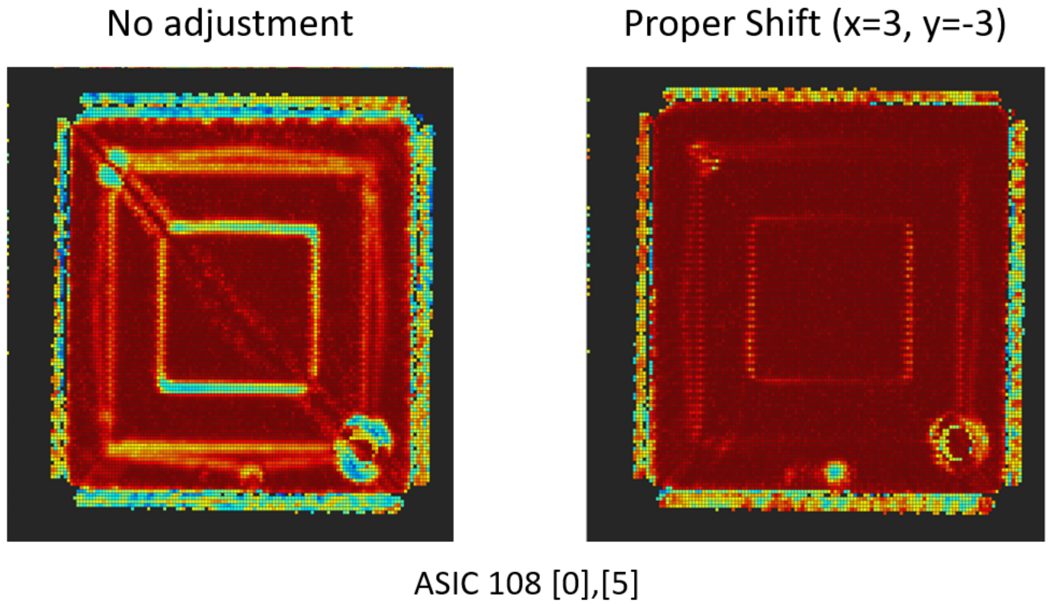

Figure 10: The difference between not translationally aligning the datasets for the correlation analysis (left) versus aligning them properly (right). Areas of yellow and blue are areas where the correlation is low. As can be seen, artificial anti-correlations are introduced and real anti-correlations of interest (bottom of the ASIC) are lost if the data is not aligned properly.

row-wise, from left to right, for a pixel of amplitude value > 0.45. Nearly every pixel point on the front surface of the ASIC contains a maximum amplitude value greater than 0.45, and thus is an excellent place to constrain the edges of the ASIC. Upon searching row-wise (in x-direction), once the algorithm finds a value greater than 0.45 the pixel location is stored in an array and the loop is broken. The code moves onto the next row and repeats the process across the entire ASIC (and each y coordinate point). The procedure is repeated column-wise to find the top-edge of the ASIC (by searching in y and keeping x fixed for each column). Two arrays of data, containing around 100 pixel location points are then averaged (after removing the largest outlier) to yield an accurate left and top edge of the ASIC that can be used as a 'slice-point' for aligning two different measurements. The averaging across 100 datapoints leads to a reliable edge-point for the analysis. Since the exact physical dimensions of the ASIC is known, the scaling ratio between pixels and millimeters is used to take the left and top edge points to cut the right and bottom edges. The result is a cutout of the ASIC containing only the relevant, internal part (and omits the pins).

The data arrays between cold cycle measurements, as discussed above, start at a different location relative to one another. After the edges are found, the data from each cold cycle can be re-loaded to properly align the two measurements. Each data is thus shifted in x and y so that their spatial positions are aligned. The correlation analysis can then be performed. To highlight the difference the alignment procedure makes in the correlation analysis, a correlation colormap showing no alignment and post-alignment is shown in Figure 10. The difference is stark. It is evident from the figure that if no alignment is enacted, then most of the anti-correlation regions are simply an artifact of a translational difference between data. Figure 10 shows the difference of a relatively small alignment shift; this effect is even more profound in other ASIC data. In addition, by properly aligning two measurements real anti-correlation regions can be observed that are otherwise hidden or suppressed, as can be seen on the bottom edge of the ASIC. The alignment procedure allows for a "hands-free", automated way to ensure the correlation differences observed are real, and not just from differences in placement between measurements. Prior to the automated alignment, this had to be done manually on an ASIC-by-ASIC basis. The alignment procedure is generally accurate to around 1-2 pixels (130-260 micrometers). Some translational differences exist in the data, even after aligning (due to

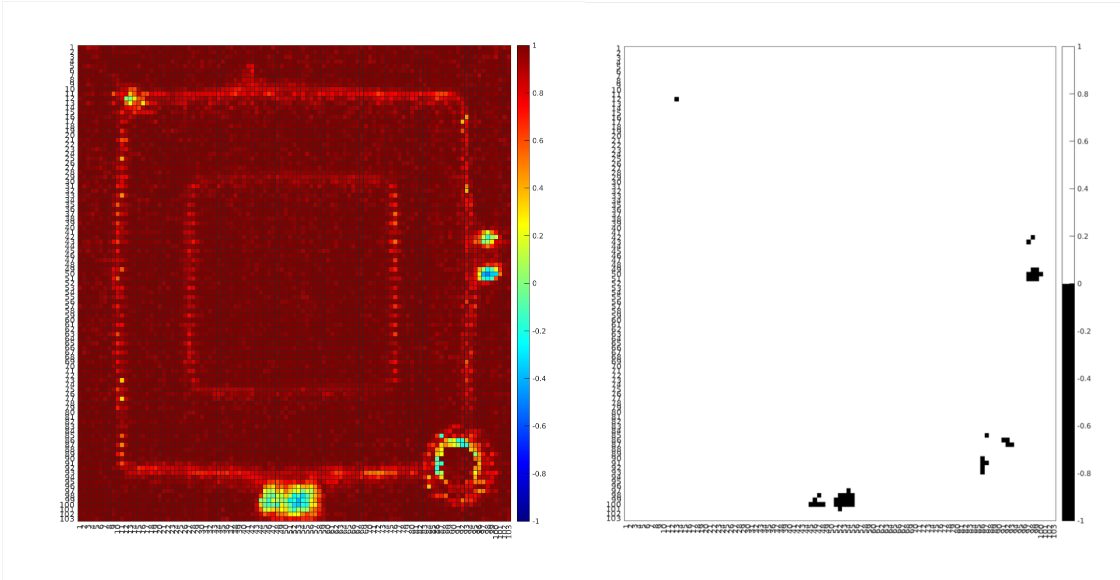

Figure 11: Output plots from the MATLAB analysis. Shown left is the correlation colormap, scaled to show just the front surface and no external pins. The negative (right) of the colormap is a threshold plot where only regions of anti-correlation (values less than 0) are shown. This plot is then used in the OpenCV analysis.

the limited resolution), but the addition of this procedure greatly improves the accuracy of the correlation analysis.

Upon loading the shifted data, a new time of flight window is specified. Before correlation is performed, one final step must be completed. Some ASICs in the data (unfortunately) were and in the future may be placed rotated by 90, 180, or 270 degrees in the fixture or placed in the wrong fixture slot relative to the initial measurements. This human-induced error must be corrected in the code, as the correlations would be either from rotational differences or from overall different ASICs and not thermally related. In this section of the code, specific rotations can be applied using the MATLAB rot90 function. In additon, the code can jump slots if needed to account for ASICs being in incorrect positions. This allows for the user to correct for any human errors during data taking.

Finally, a correlation analysis is performed for each spatial point along the ASIC using MATLAB's xcorr function. The correlation corresponds to whether a range of the waveform (in time) is in phase. Waveforms that are out of phase relative to one another will yield low correlation values. A low-correlation value corresponds to a difference in the ASICs acoustic impedance within the specified depth range, at the specified grid point. The correlation values (one for each gridpoint) are stored as a 2D array, which is then plotted as a correlation colormap. The finalized output is shown in Figure 11. The color axis for the colormaps is continuous, where bluer regions are places where the correlation is negative (anti-correlated), and redder regions are places of high, positive correlation. The axis runs from -1 to 1, as do the correlation values. In addition, a negative colormap is produced with a threshold color axis. Any value of correlation less than 0 (any negatively correlated pixel points) are represented as regions of black. The rest is white. These negative correlation threshold plots are used in the next stage of the SAM analysis. The negative correlation colormap is also shown in Figure 11.

5 OpenCV Analysis - Quantifying Regions of Anti-Correlation

The next step of the SAM correlation analysis requires quantifying the regions of anti-correlation. The MATLAB analysis detailed above produces correlation colormaps that are beneficial in identifying areas within the ASIC that have changed due to thermal-related stress. By producing a negative, threshold correlation plot, these regions are easily identifiable and can be analyzed using an image-based algorithm. For this analysis, an image AI software called OpenCV is used. OpenCV is a wide-reaching, optimized computer vision library, containing tools and hardware for analyzing images. The software is excellent at identifying shapes, figures, and objects of interest from backgrounds within image files. The extension of Python OpenCV to this project was straightforward, and well within the intended use of the software. In short, all the software does is pick out black shapes on a white background. The software was recommended by a fellow Physics graduate student at ISU, who uses the algorithm for Biophysics related research.

The description of the OpenCV analysis in Python follows the code **ACR OpenCV Analysis.ipynb**. The OpenCV software used in this analysis requires image filetypes to work, therefore all MATLAB output plots are .PNG filetypes. To aid the algorithm in correctly identifying regions of anti-correlation, the images must be cropped to only contain relevant areas to the analysis. Thankfully, the MATLAB output plots are framed such that the outer edge of the ASIC is the border of the correlation colormap. After loading in the necessary python package for OpenCV (cv2), each ASIC correlation plot produced from the MATLAB analysis is loaded in using cv2.imread. The loaded image is cropped such that the x and y axes, as well as the color axis are excluded from analysis. In extending this analysis to other chips, manual calibration of the exact cropping of the images is required. The only specification is that the image fed to the algorithm only contains relevant areas to be analyzed. In addition, the reader is cautioned that different computers may produce slightly different dimensioned MATLAB output plots. This may require some fine tuning of the cropping parameters utilized in the Python code. Manual adjustments are easy to do and are not a significant barrier to extending this analysis. A new image is then written with the correct cropped dimensions using cv2.imwrite, then re-loaded. A negative is produced of the image using cv2.cvtColor. This step is not necessary as the MATLAB analysis already produces a negative image but is left in the code in case other correlation plots with color are used. Gaussian blurring is done using cv2.GaussianBlur to reduce image noise, then image thresholding is applied using cv2.threshold to apply a uniform color for each of the above-threshold pixels. This is done to aid the algorithm, but is not necessary if a negative colormap is fed into the code. The threshold, negative, cropped image is re-written, then re-loaded into the software, before finally performing the image identification. The specific image identification utilized is a contour-finding algorithm, cv2.findContours. OpenCV identifies the contours of the anti-correlation regions, then stores each contour's center of mass location (in x and y) and area to an python array. The center of mass is found through a moments calculation, where a constant density is assumed. OpenCV produces composite images containing the contours of the anti-correlation regions, as well as their centerpoints. These are used to double check the moments calculation if needed. Across all ASICs, the moments calculation of the centerpoints align with an intuitive expectation. Most of the contours contain only a few pixels, so their centers are relatively meaningless. Only for the larger contours are the centerpoint locations subject to potential errors. Though information is not known about the specific errors of the moments calculation, it can be assumed to be on order of a pixel size, or around 130 micrometers. An example of a contour for an ASIC correlation plot identifid with the OpenCV framework, as well as the calculated centerpoints is shown in Figure 12. Figure 12 shows the same ASIC as Figure 11.

More information about OpenCV can be found at https://opencv.org. After the code is ran for all ASICs, further analysis can now be perfomed.

The array ACR in the code contains three initial dimensions; the first and second are the x and y pixel coordinate locations of each anti-correlation region for each ASIC. The third dimension is the total area of the contours, given in image pixels squared. It is important to note that the coordinates are given in pixel values. Since the exact dimensions of the ASIC is known (in millimeters), and the image is cropped such that only the ASIC front wall is framed, it is easy (and done in later kernals within the code) to convert to physical units to identify the exact locations of the anti-correlation regions. Multiple anti-correlation regions

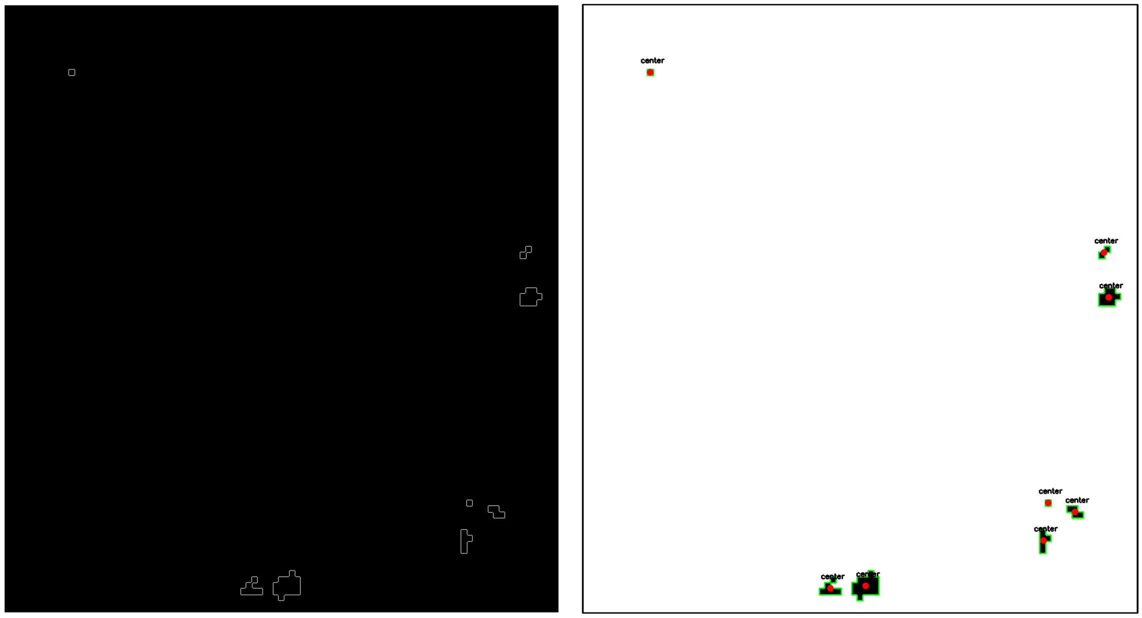

Figure 12: Anti-correlation region identification using the OpenCV contour finding. The ASIC used for these plots are the same as in Figure 11. The plot on the right shows the centerpoints and edges of each contour, which are used in quantifying the anti-correlation regions.

(contours) can be identified for a given ASIC, and thus each dimension of the ACR array contains all ASICs and their associated anti-correlation regions.

Further data processing is required before quantifying the regions of anti-correlation. As can be seen in Figure 11, the lower right and upper left areas on the ASIC display regions of low-correlation. These specific features on the ASIC are from acoustic scattering on circular grooves cut into the front wall of the LArASIC. Regions like this must be removed, as they do not contain correlations related to thermal stress. This next paragraph details the specific noise reduction and filtering used in the Stress Test study for the LArASICs, like the removal of scattering sites from circular grooves, but serves as a general explanation for how to remove unwanted regions of data using this framework.

The circular groove sites must be identified and removed. The areas of the groove sites, and other areas of data filtering can be identified by plotting an example correlation image in the python package Matplotlib.pyplot. Matplotlib.pyplot has a built in cursor showing its coordinates - this can be used to record the outline of regions that are to be omitted from the data. In addition to the removal of the circular groove sites, five of the 100 total ASICs are not included due to data corruption issues as well as alignment issues. If this data was included, it would produce many, irrelevant anti-correlation regions. Additional data filtering can be performed if needed. For this specific analysis, internal regions of low-correlation were found to be overwhelming due to small alignment differences in the ground plane between measurements, as well as single pixel points that were missing data. As a result, the internal region (away from the edges) were omitted. Differences in the distribution of anti-correlation regions before and after data subtraction should be carefully analyzed to ensure bias is not being introduced into the results. As can be seen in the code, more detailed analysis can be performed by filtering ASICs based on failure modes and anti-correlation regions based on proximity to different pin types. ASIC and other electronic datasheets are likely required to localize regions of anti-correlation to specific pins and wire bond locations. Area conversions can be performed from pixels squared to real units, since the dimensions of the ASIC in millimeters is known. Statistics and hypothesis testing are also conducted and several different distributions and plots can be created. More detail can be found in the embedded comments. Some distributions and plots produced in the Stress Test Study are shown in Figures 13 and 14 as examples of the power of this data processing framework.

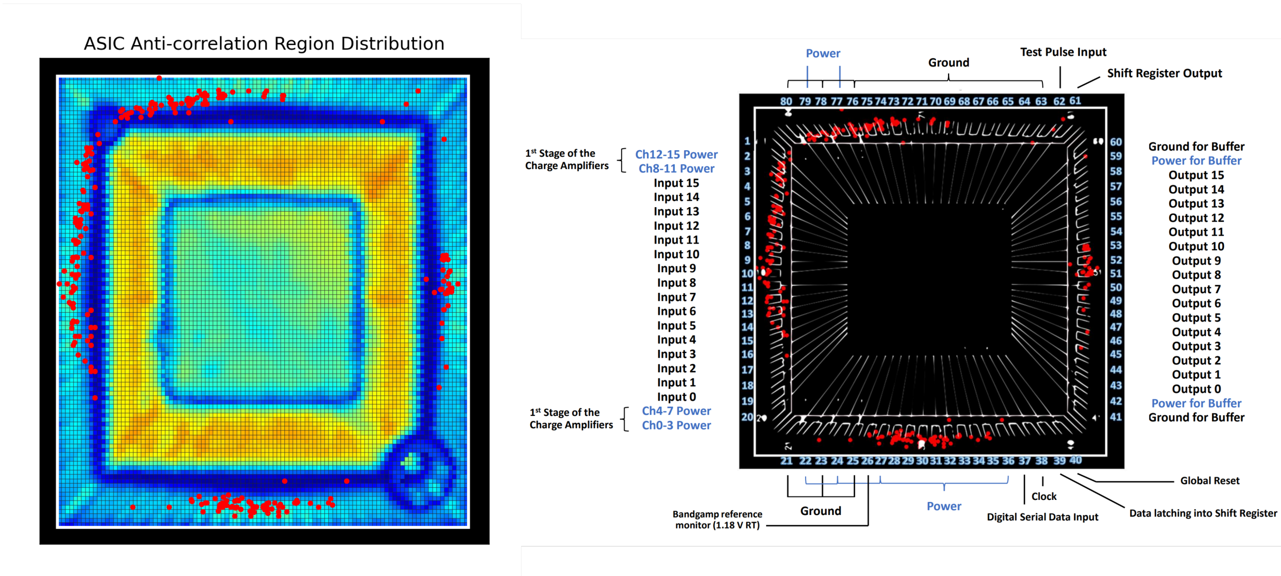

Figure 13

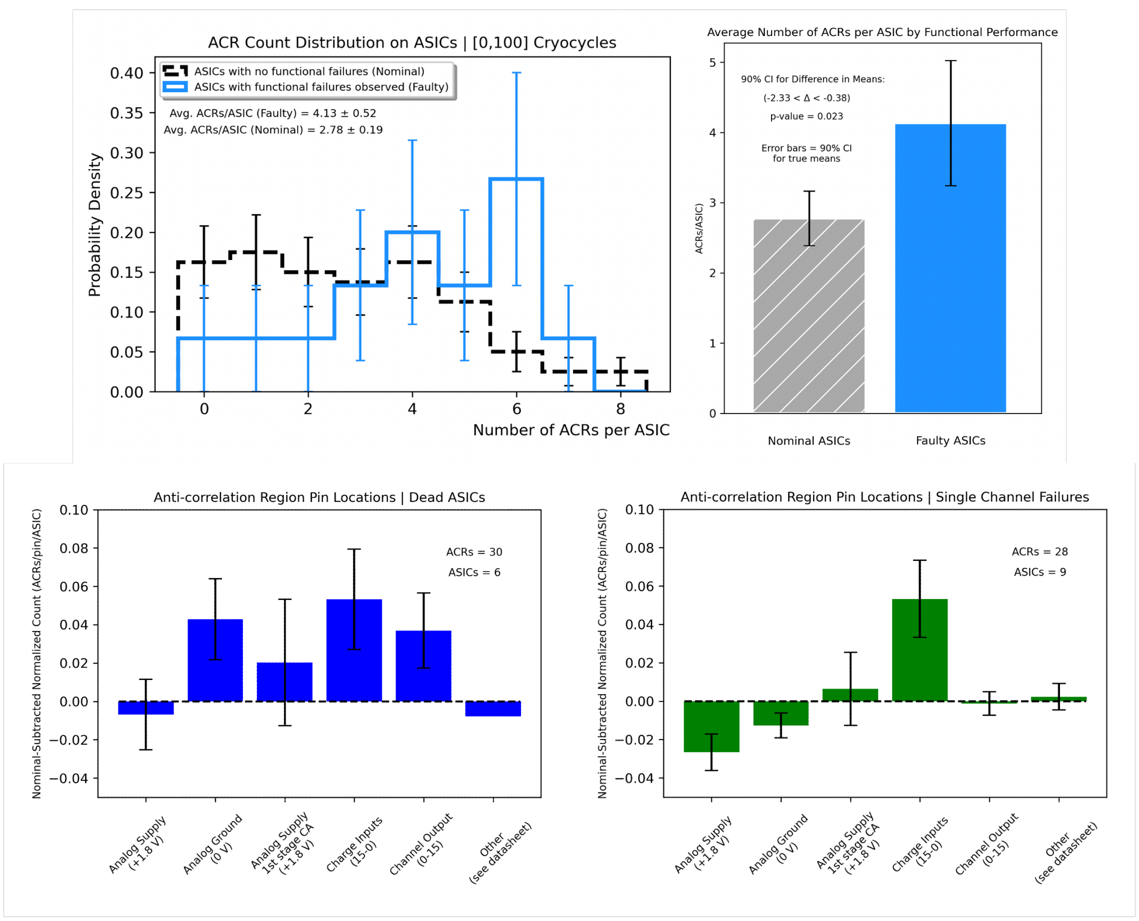

Figure 14