Internship at L3Harris: Software Engineering and Signal Processing

CS4991 Capstone Report, 2024

Natasha Mathew
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
nm2jdj@virginia.edu

ABSTRACT
L3 Harris, an American technology company and defense contractor, found that its sonar testing process was outdated and needed additional capabilities. In order to address this problem during my internship, I worked on beamforming and pre-processing techniques for the passive sonar of the Multi-function Towed Arrays (MFTA). I worked on existing C programs that performed beamforming and added capabilities for half azimuth, an array with hydrophones that cover 180 degrees. The problem our team ran into was the half azimuth beams were not compatible with the functions associated with the active sonar that built off the array I was working on. We created a function that could mirror the beams of the half azimuth before the tap off to the active sonar. I added a new select_mode for the beams in the configuration file called beam_mirror, which loops through the array of half-azimuth beams and copies the beams over to a new side, creating a full-azimuth array. Creating this function allowed for a more efficient, seamless beamformer whose function was validated through testing with MATLAB comparison tools. In the future, there should be more thorough testing of the new beam_mirror function and of the passive and active sonar processes by observing the total actual output of the array compared to the expected in order to ensure the functionality of the beamformer.

1. INTRODUCTION

Beamforming is a signal processing technique used in sensor arrays to filter and process signals received. Submarines utilize beamforming capabilities through their towed arrays, an arrangement of hydrophones on a cable that is dragged behind the submarine. These hydrophones can detect different sounds and signals underwater, enabling the submarine to have a better understanding of threats and their surroundings. One type of sonar that submarines can use in order to detect the sounds around them is passive, which involves a submarine listening for sounds, rather than emitting sounds and listening for the echoes.

Beamforming is essential in maritime operations in order for submarines to be able to detect and track underwater objects. Passive sonar is also extremely important for submarines to maintain their stealth in critical situations. Through utilizing beamforming and passive sonar, submarines can maintain situational awareness as well as safety.

2. RELATED WORKS

Passive sonar is important for stealthy listening and is utilized in defense and scientific application. A computer system can analyze data received and identify objects such as vessels, as well as speed of ship or types of weapons (Morin, 2019). Hydrophones are attached to a towed array for passive sonar and are kilometers away, which improves the signal to noise ratio. The towed array sonar is effective for detecting faint sounds such as the low noise-emitting submarine threats or seismic signals.

The way the hydrophones process the passive sonar it receives is through beamforming, which is a key aspect of signal processing computation. There are different methods to beamforming such as CBF, which is a traditional method; or ABF, which is an adaptive beamforming method. With the CBF method, because the angle resolution is limited, if the targets are within the main lobes of the beams then it cannot be distinguished (Yin & Chen, 2023).

The ABF method is more effective in general because it utilizes different weight vectors by adjusting each array element in different signal environments. This method uses weighted filtering in order to enhance the signal and suppress interference signals, which improves the overall sonar performance.

3. PROJECT DESIGN

During my internship at L3 Harris, I worked on a signal processing team. My project was working on the beamformer for the MFTA towed array. More specifically, I worked on signal processing techniques for the passive sonar of the MFTA towed arrays. The main task throughout my internship was improving the beamforming capabilities of the array to complement the new software of the company. I had to convert their legacy code into SPE in order to have more current beamforming abilities. I helped improve the beamforming capabilities of the array through the tests and functions that I developed, such as beam_mirroring for the beamformer. Improving the functionality of the beamformer was significant for the submarines using the MFTA towed arrays to get the most accurate passive sonar readings of their surroundings.

The code that I developed for the beamformer was written in C in a linux environnement. I had to transfer the legacy code into the new SPE code; however, sometimes it did not translate directly and I had to write helper functions. The main issue I ran into was turning the half azimuth beams that the beamformer was inputting and outputting them as full azimuth beams. This was crucial in being able to go from the passive sonar ports to the active sonar in the code. My solution was creating a beam_mirror function that I could add as a select_mode configuration for a specific beamformer. One thing I had to take into account when creating this function was the guard beams and the endfire beams.

There are two types of guard beams: one type where there is one at the beginning and one at the end; and the other type where there are up to six only at the end. Endfire beams are beams at 0 degrees and 180 degrees that act as placeholder beams. When I encountered an endfire beam, I had to make sure to not mirror it, and just skip over it. When I encountered a guard beam, I had to make sure not to mirror it, but rather move them to the end of the fully mirrored beams. In the code that I developed, the beams were represented as elements of an array. So in order to mirror beams, I doubled the size of the array and took into account the number of endfire and guard beams, to get the right size array. Then I looped through the array starting at index 0, and copied the beam data from that index into its corresponding

mirrored position. I moved the guard beams first to the end in order to properly account for them before I began mirroring the rest of the standard beams.

As I developed this code, I was running tests in order to ensure that the proper beam data was getting copied over, and that the correct number of elements were produced in the output. I created test files for the different cases of beam data to test my function through developement. As I developed code, I would also run raw beam data through my code and output the results in a test file. Then I would take the test files and the expected output files of the raw beam data and put them into Matlab comparison tools. Matlab allowed me to transfer the raw data from both the legacy code files and the new updated SPE files and put them into readable graphs that enabled me to see the discrepancies.

4. RESULTS

When I ran my code through the Matlab comparison tools, I found minimal differences between the legacy output and the new SPE code that I wrote. This demonstrated that the new beam_mirror function I wrote worked as intended. Now that the passive sonar beamformer is working with the new SPE code, the beam_mirror function can be utilized in the other beamformer code that the team works on in order to start updating the old legacy code.

5. CONCLUSION

This project is important because it improved the beamforming capabilities of the MFTA towed array. By creating a new select_mode, I expanded the capabilities of the different processes that can run through the beamformer. Creating a new function allowed me to successfully update the old GMP code to the SPE. Through updating the old software, I was able to help the process of the transition to the new updated SPE, which will make writing code more efficient and allow for outputting more technologically advanced deliverables to the clients.

6. FUTURE WORK

More detailed tests-should be done with real beam data in order to ensure the functionality and to validate the results. I would run tests with real sonar data in order to ensure that the way that the beamformer processes it is accurate. I would validate results with more in depth Matlab comparison tools as well.

Overall, the code that I wrote improved the functionality of the beamformer, and the new function that I created can be utilized in future updates.

REFERENCES

Morin, S. (2019). *Towed Array Sonar: Uses and Design Challenges*. Ametek SCP; Ametek SCP. https://www.ametekscp.com/knowledge/blog/2019/march/towed-array-sonar-uses-and-design-challenges

Yin, B., & Chen, J. (2023). *Application of adaptive beamforming technology in underwater acoustic*. Ieee. IEEE. https://ieeexplore.ieee.org/document/10105792