SCIENCE AND TECHNOLOGY ORGANIZATION
CENTRE FOR MARITIME RESEARCH AND EXPERIMENTATION

Reprint Series **CMRE-PR-2019-127**

Real-time continuous active sonar processing

Gaetano Canepa, Andrea Munafò, Stefan Murphy,
Michele Micheli, Luca Morlando

June 2019

Originally published in:

OCEANS 2015, 18-21 May 2015, Genoa, Italy,
doi: 10.1109/OCEANS-Genova.2015.7271658

About CMRE

The Centre for Maritime Research and Experimentation (CMRE) is a world-class NATO scientific research and experimentation facility located in La Spezia, Italy.

The CMRE was established by the North Atlantic Council on 1 July 2012 as part of the NATO Science & Technology Organization. The CMRE and its predecessors have served NATO for over 50 years as the SACLANT Anti-Submarine Warfare Centre, SACLANT Undersea Research Centre, NATO Undersea Research Centre (NURC) and now as part of the Science & Technology Organization.

CMRE conducts state-of-the-art scientific research and experimentation ranging from concept development to prototype demonstration in an operational environment and has produced leaders in ocean science, modelling and simulation, acoustics and other disciplines, as well as producing critical results and understanding that have been built into the operational concepts of NATO and the nations.

CMRE conducts hands-on scientific and engineering research for the direct benefit of its NATO Customers. It operates two research vessels that enable science and technology solutions to be explored and exploited at sea. The largest of these vessels, the NRV Alliance, is a global class vessel that is acoustically extremely quiet.

CMRE is a leading example of enabling nations to work more effectively and efficiently together by prioritizing national needs, focusing on research and technology challenges, both in and out of the maritime environment, through the collective Power of its world-class scientists, engineers, and specialized laboratories in collaboration with the many partners in and out of the scientific domain.

Copyright © IEEE, 2015. NATO member nations have unlimited rights to use, modify, reproduce, release, perform, display or disclose these materials, and to authorize others to do so for government purposes. Any reproductions marked with this legend must also reproduce these markings. All other rights and uses except those permitted by copyright law are reserved by the copyright owner.

NOTE: The CMRE Reprint series reprints papers and articles published by CMRE authors in the open literature as an effort to widely disseminate CMRE products. Users are encouraged to cite the original article where possible.

Real-time continuous active sonar processing

Gaetano Canepa
Andrea Munafò
Michele Micheli
and Luca Morlando
NATO STO Centre for Maritime Research and Experimentation
19134 La Spezia, Italia
Email: {gaetano.canepa,andrea.munafo,
michele.micheli,luca.morlando}@cmre.nato.int

Stefan Murphy
Defence Research & Development Canada
Atlantic Research Centre
Halifax, NS, Canada, B3J 1G2
Email: murphy.stefan@gmail.com

***Abstract*—This work describes the development of continuous active sonar (CAS) processing at CMRE. The software uses sub-band processing to achieve a faster update rate than is possible with pulsed active sonar (PAS). The software development was based on CMRE's PAS processing software, `CAINPro`, which has been thoroughly tested during previous sea trials and in post-processing data analysis. Computational efficiency was carefully considered and many optimizations were made so that the software can run in real-time using the constrained computing resources on board CMRE's Ocean Explorer autonomous underwater vehicles (AUV). The software was successfully tested during the REP14 Atlantic sea trial in July 2014, and was able to demonstrate real-time detection of an echo repeater on all nine sub-bands that were processed. The CAS algorithm runs in real-time on the processing board installed on the AUV.**

I. INTRODUCTION

In the last few years there has been a growing interest in the concept of Continuous Active Sonar (CAS) for Anti-Submarine Warfare (ASW). This is largely due to the advances in sonar hardware and computing speed which now allow the CAS concept to be implemented in the field. The concept is to transmit a waveform continuously. Typically this means repeating the transmission of a long-duration waveform, without pause, with a repetition rate similar to that used with traditional pulsed active sonar (PAS). In PAS systems, a short continuous wave (CW) or frequency modulated (FM) pulse is transmitted, followed by a relatively long period of time during which no transmission occurs and the receiver listens for echoes. In contrast, CAS systems transmit and receive echoes simultaneously, and the resulting continuous target insonification can potentially improve ASW detection and tracking performance.

CAS can potentially improve ASW sonar performance in two ways. First, the total transmitted energy can be increased by extending pulse duration with a constant source level, and this can increase target detection range. Pulse energy is maximized with a 100% duty cycle; however, transducers may not be able to achieve this performance in practice. Additionally, the ideal processing gain may not be achieved for high time-bandwidth product waveforms, especially in the littorals where sound propagation is complex. Rather than processing the entire pulse coherently, suitable waveforms such as linear FM (LFM) sweeps can be segmented and treated as a series of short, non-interfering pulses, which are processed individually. This type of sub-band processing is the basis of the second performance improvement that CAS offers: increasing the update rate of sonar contacts while maintaining the same pulse repetition interval and corresponding search radius. Some additional advantages of CAS have been discussed in [1].

It is important to highlight that the potential advantages of CAS have not yet been scientifically validated through experiments. This validation is especially important for littoral waters where sound propagation is complex and false alarms can overwhelm active sonar systems. CMRE has developed a real-time CAS processing chain to work with their research sonar systems, allowing immediate feedback of CAS performance during upcoming internal and JRP sea trials. This real-time feedback is critical to ensure systems are working properly and data is being collected that is suitable for thorough scientific analysis in post processing.

This report provides the details of CMRE's CAS processing, which is based on software developed and validated at CMRE for PAS systems [2]. Computational efficiency was an important consideration in the development since an immediate goal was to run the processing in real-time on board CMRE's Ocean Explorer (OEX) autonomous underwater vehicles (AUVs) with constrained computing resources. The software was tested for the first time during the REP14 Atlantic sea trial that was held off the coast of Lisbon, Portugal, in July 2014. The software ran in real time on the AUVs, and was able to successfully detect and track an echo repeater towed by NRV Alliance in a bistatic configuration. The CAS pulse, an 18 s LFM swept over the 1800–2700 Hz band, was broken into nine sub-bands. This provided detections on the echo repeater at a rate nine times faster than would be possible with pulsed sonar.

The document is organized as follows: Section II describes the technical details of the CAS processing components; Section III shows preliminary experimental results obtained with the CAS processing chain during the REP14 Atlantic sea trial; and conclusions are summarized in Section IV.

II. CMRE SONAR PROCESSING SOFTWARE

The implementation of CMRE's CAS signal processing algorithm is called `CAS-CAINPro` and its development in C++ was based on an algorithm called `CAINPro` that was previously developed by CMRE for PAS processing [2]. The algorithm uses sub-band processing as discussed in Section I. This method breaks up a long pulse into short segments that are processed individually using PAS processing. The approach of building from the `CAINPro` PAS processing was therefore suitable. `CAS-CAINPro` includes code optimization for computation speed and quality of output, and the details of these algorithm changes are also presented. `CAINPro` and `CAS-CAINPro` are libraries written in the C++ programming language.

A. *CAINPro PAS processing software*

The `CAINPro` program include the following processing steps, also shown in Figure 1.a:

1) Data from the array elements are read from a file and converted to double-precision values. The mean is subtracted from the data.
2) Array data are demodulated to baseband in the time domain (TD).
3) Array data are transformed to the frequency domain (FD) using the Fast Fourier Transform (FFT).
4) A large-bandwidth beamformer in the frequency domain is applied to obtain beamformed data.
5) Beamformed data are matched filtered with the FFT of the demodulated replica of the transmitted signal.
6) The inverse FFT (IFFT) is used to transform the data back to the time domain, yielding a complex envelope
7) Matched-filtered, beamformed data are downsampled and the magnitude of the envelope kept (phase discarded)
8) Median or split-window normalization is applied to turn the envelope into a random variable with $\sigma = 1$.
9) A detection and clustering algorithm is applied and the top ranked clusters, up to a given maximum number, are taken to form sonar contacts.

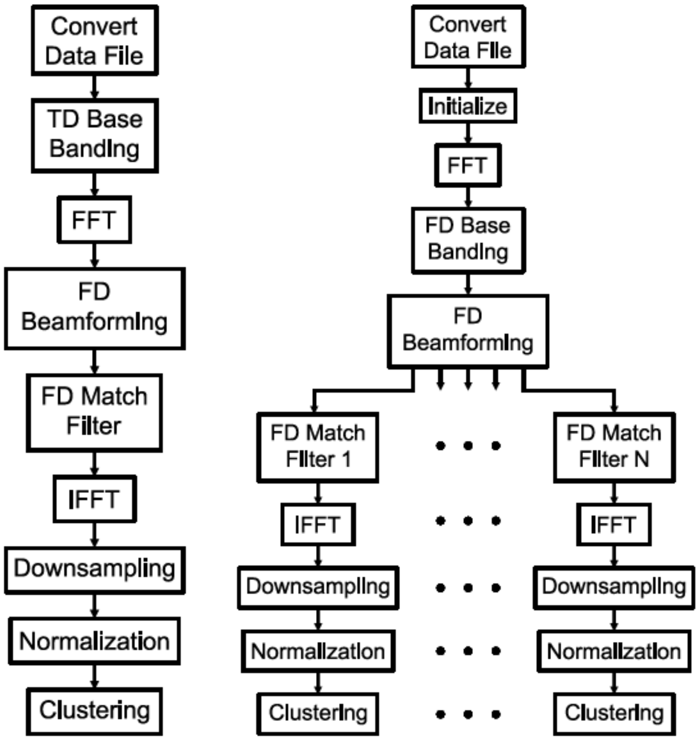
The original `CAINPro` program was used to test sub-band matched filtering. In the test case an 18 s LFM pulse was divided into nine, 2 s sub-bands. Each sub-band was then used as a replica and the full `CAINPro` processing chain was applied.

An intermediate software, called `cas-CAINPro`, noting the lower case prefix, was initially made to validate some of the code changes. To ensure the changes to the processing stream produced identical results, the output from `CAINPro` and `cas-CAINPro` were compared. A bit-to-bit match verified the outputs were identical. The processing speed of `cas-CAINPro` was not adequate, however. A number of algorithm optimizations were required to allow real-time processing, and these are now presented.

B. *CAS-CAINPro CAS processing software*

The `CAS-CAINPro` program includes a number of improvements to `cas-CAINPro` that make it possible to run in real-time, for example, on board the OEX AUVs, which currently have an AMD quad-core payload computers with 8 GB RAM.

The first important step in increasing the efficiency of the software was to change the structure to consolidate all instances of *initialization* as shown in Figure 1.c. This made reading, debugging, modifying, and timing the code much easier.



(a) (b)

Fig. 1. Block diagrams summarizing how (a) `CAINPro` and (c) `CAS-CAINPro` process sonar data in the time domain (TD) and frequency domain (FD) to obtain clustered detections, which form sonar contacts. (c) includes a consolidated step for initialization and the newly developed frequency domain baseband algorithm.

Vectorization was also employed to increase computational speed. Modern central processing units (CPUs), including those on board the OEX AUVs, use registers. Registers are temporary storage blocks of 256 or 512 bits in modern architectures. Vectorization is a practice that maximizes the parallelization resulting from the use of large registers. It can increase speed by a factor of eight for single precision variables and four for double precision variables. Vectorization can be implemented in the C language for simple functions like summation or multiplication of two arrays.

Multi-threading was another way to achieve gains in processing speed. Software can be developed to take advantage of multi-core CPUs, which are found in most new computers or single board computers (SBC). Multi-core CPUs can also be more energy efficient, which is an important factor for the OEX AUVs which have limited battery power. It is possible to transform single-core code to multi-core code with relatively small effort using the OpenMP® Application Programming Interfac (API).

Figure II.b shows the block diagram of the final `CAS-CAINPro` program. The steps are summarized below with the improvements made included in brackets:

1) Data are read and converted to double-precision
2) Initialization
3) Array data are transformed to the frequency domain

4) Array data are transformed to baseband
5) A large bandwidth beamformer in the frequency domain is applied to obtain beamformed data
6) Beamformed data are matched filtered and downsampled
7) Matched-filtered beamformed data are downsampled
8) Normalization is applied to return random variable with $\sigma = 1$
9) A detection and clustering algorithm is applied and the top ranked clusters are taken up to a defined maximum number

Further details for each step are included in the following sections.

1) Data reading, conversion and de-trending: The conversion of acoustic data from integer to floating point format is performed using vectorized routines. It is no longer necessary to remove the mean from the data because a new baseband algorithm was developed that operates in the frequency domain (see Section II-B4).

2) Initialization phase: The optimization of this phase is not strictly necessary for real-time processing; however, since the `CAS-CAINPro` library is also used for post processing, reducing the initialization time was worthwhile. Most of the improvement was obtained by calculating the beamformer coefficients (see Section II-B5) using a public domain library that vectorizes the calculation of the complex exponential [3].

3) FFT library settings: All of the FFT and IFFT functions used in `CAS-CAINPro` are calculated using the `FFTW` library [4], a standard public domain library included in the `CAINPro` library. While `CAINPro` only uses FFT lengths that are integer powers of two, `CAS-CAINPro` uses FFT lengths that are powers and multiples of two, three, five, seven, eleven and thirteen. The algorithms that use lengths other than powers of two are marginally less efficient than the classical Cooley-Tukey FFT algorithm, but the reduction in FFT length offsets the efficiency loss and reduces calculation time in the following steps of the algorithm.

4) Baseband algorithm: A baseband algorithm is applied to the data to allow a reduction of the sample rate, which reduces computation time of subsequent algorithms. A signal at baseband maintains all of the information in the signal band, but is shifted in frequency so that the sampling rate can be reduced to the bandwidth of the complex signal rather than twice the highest frequency contained in the real signal.

For the wide-band low frequency waveforms typical of CAS, the downsampling factor is usually below 10. In this case frequency-domain baseband algorithms are more efficient, so the original time-domain algorithm was replaced with a frequency-domain implementation.

5) Beamforming in the frequency domain: A time-domain beamformer operates by delaying the signal recorded on each element of an array and summing the signals from the elements together. The delay applied to each element is calculated from the angle for which a beam is to be steered. A shading function is also applied to reduce side lobes.

The existing beamformer was optimized for `CAS-CAINPro` to exploit vectorization and multi-threading. The data are kept in the frequency domain after beamforming because the matched filter is also implemented in the frequency domain.

6) Matched filtering, downsampling, normalization and clustering: This parts of the algorithm saw minor optimizations consisting in introducing vectorization multithreading wher possible.

C. Computing on board OEX AUVs

The CAS signal processing chain just described was implemented on CMRE's OEX AUVs and on board NRV Alliance. The improvements to computational efficiency allowed the software to run in real-time using the constrained computing resources on board the AUVs. The payload section of each AUV contains two computer systems: one for data acquisition from the SLITA acoustic array towed by the AUV, and one for signal processing as well as autonomous decisions through MOOS [5], [6] (Mission Oriented Operating Suite). The slita-pc is a PC/104 with a single-core 1.4 GHz Pentium-M processor and 1 GB of RAM. The backseat-pc is an AMD quad-core 2 GHz processor with 8 GB of RAM. Both computers run Linux operating systems and are connected via gigabit Ethernet.

III. EXPERIMENTAL RESULTS

The newly developed `CAS-CAINPro` CAS processing software was tested by comparing it to the older, well-tested `CAINPro` software, and by confirming proper contact formation using CAS data from the REP14 Atlantic sea trial. During the sea trial an 18 s, 1800–2500 Hz LFM, was transmitted using the mid-frequency ATLAS source towed by NRV Alliance. The `CAS-CAINPro` software is designed to run on the OEX AUVs in multistatic configuration, but for the test results presented in this report the effectively monostatic setup of the ATLAS array was used. The echo repeater simulated a target by adding a 4 s delay before repeating signals. A special algorithm was used to handle the near-continuous waveforms used. Ideally the echo repeater would have been towed by a second ship to provide a more realistic target simulation; however, this was not possible during REP14 Atlantic.

The CAS processing software was configured to run on the 24 s data files acquired from the ATLAS array. The 18 s LFM transmission was segmented into 2 s sub-bands for processing. The nine sub-bands did not overlap and had a bandwidth of 77.8 Hz. The processing chain described in the previous sections was executed, and the final stage of detection and clustering used a detection threshold of 6 dB. This relatively low threshold produced a high number of false alarms, but only the top twenty highest signal-to-noise ratio (SNR) contacts were kept by the software (this is tunable parameter).

A. Output comparision

The first test of `CAS-CAINPro` was a comparison with `CAINPro` algorithms to confirm the outputs were the same. All of the outputs of the algorithm components matched within error comparable to double-precision floating-point error.

B. Improvement to computing time

Next, the improvement to computation time was measured. Table I shows the computation times for CAS processing using `CAINPro` and `CAS-CAINPro`. Recall that the original `CAINPro` software was designed for PAS processing. The computation time for some steps was therefore multiplied by the number of sub-bands, in this case nine, to estimate the time required to process all of the sub-bands. Values that include this multiplication factor are marked with an asterisk in Table I.

The timing measurements in the first two columns of Table I were obtained by running the software on a desktop computer with an Intel® Xeon® CPU E5-2620 v2 2.10 GHz using two cores for multi-threading. The last column shows the processing time for `CAS-CAINPro` on the computer system on board the AUVs.

Computation times measured using `CAS-CAINPro` were dramatically improved from those measured using `CAINPro`. Without considering initialization, the processing time for nine sub-bands was 6.69 s for the original `CAINPro` and only 2.20 s for the `CAS-CAINPro`, a speed up of approximately three times.

The processing time on the AUV computer is increased to 4.23 s; however, it is important to recall that these tests were performed using ATLAS array data. During sea trials, the AUVs process data from the SLITA array, which has half the number of elements, only forms half the number of beams. The `CAS-CAINPro` algorithm is not only faster but also reduces computation noise introduced by the original `CAINPro` algorithm. Ultimately, the speed up in processing time will allow real-time processing of CAS signals with larger bandwidth and using a larger number of sub-bands, noting that processing time must be less than the pulse repetition interval for real-time operation.

C. Testing with REP14 Atlantic data

For the last evaluation, real data from REP14 Atlantic was processed with `CAS-CAINPro` to confirm that sonar contacts were being formed as expected.

Figure 2 shows the spectrogram of a single element (number 7) for a 24 s ATLAS data file. The spectrogram shows the direct blast from 0–18 s sweeping over 1800–2500 Hz. The lower level echo repeater signal can also be observed with a delay of 4 s from the direct blast. The CAS software produced 180 contacts for this data file, recalling that the twenty highest SNR contacts for each of the nine sub-bands are taken, yielding 180 total contacts. The contacts are overlaid on the spectrogram in Figure 2. The contacts are placed according to the start frequency of the sub-band and the time delay of the detection, thus there are nine rows of twenty contacts. The magenta, diamond-shaped contacts correspond to the direct blast, and the orange, square-shaped contacts correspond to the echo repeater. There are nine contacts formed for both the direct and echo repeater arrivals, thus a detection is made in every sub-band for both cases. This confirms the software is functioning properly. The remaining green, cross-shaped contacts represent false alarms, in some cases caused by a complication with the transmitted CAS waveform.

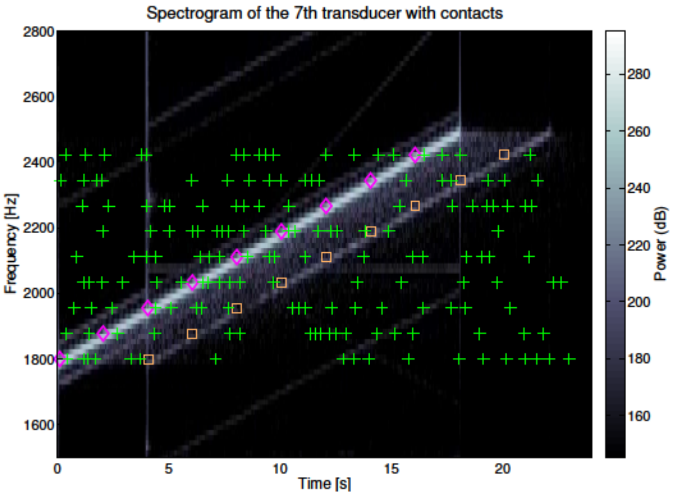


Fig. 2. Spectrogram of a 24 s file containing raw data from a single element of the ATLAS array. The overlaid symbols show contacts produced by `CAS-CAINPro` and mapped back to the raw data spectrogram using the sub-band start frequency and the contact time delay. The magenta (diamond-shaped) contacts correspond to the direct blast, the orange (square-shaped) contacts correspond to the echo repeater with 4 s delay from the direct blast, and the remaining green (cross-shaped) contacts correspond to false alarms as well as apparent echoes caused by amplifier distortion on the transmitted waveform.

The CAS waveform was found to contain unexpected distortion during the REP14 Atlantic sea trial. This can be observed in Figure 2 as LFMs slightly upshifted and downshifted from the direct blast, starting at the same time as the direct blast at 0 s and sweeping over approximately 1850–2550 Hz and 1750–2450 Hz, respectively. These effects were caused by an amplitude modulation in the amplifier used in REP14 Atlantic. The resulting sweeps have the same rate as the CAS LFM and therefore resemble an echo appearing just after the direct blast (close range contact) and just before the direct blast (long range contact from previous ping). These apparent echoes caused by the amplifier distortion are also detected by the CAS processing algorithm, as can be observed in Fig. 2. The amplifier used in REP14 Atlantic will be replaced in future trials to avoid this problem.

The previous analysis only considered the time delay of the contacts, so additional analysis was performed to examine the contact bearing estimates formed by `CAS-CAINPro`. Figure 3.a shows the beamformed, matched-filter, normalized data for the second, 1878–1956 Hz sub-band. Note that the second sub-band is more suitable for illustration than the first sub-band because the direct blast for the first sub-band coincides with edge of the plot at 0 s. The amplitude of the normalized data is mapped to the color scale, and each vertical slice represents the time series of a single beam whose angle is indicated on the horizontal axis. The contacts formed by `CAS-CAINPro` algorithm for the second sub-band are overlaid as square boxes containing the contact number, which is ordered by decreasing SNR. Contact # 1 is therefore the strongest and corresponds to the direct arrival, which is marked by the magenta box. Contact # 3 is the echo repeater arrival, marked by the orange box.

The bearing estimates should be close to 0° for the direct arrival and echo repeater arrival because the ATLAS array,

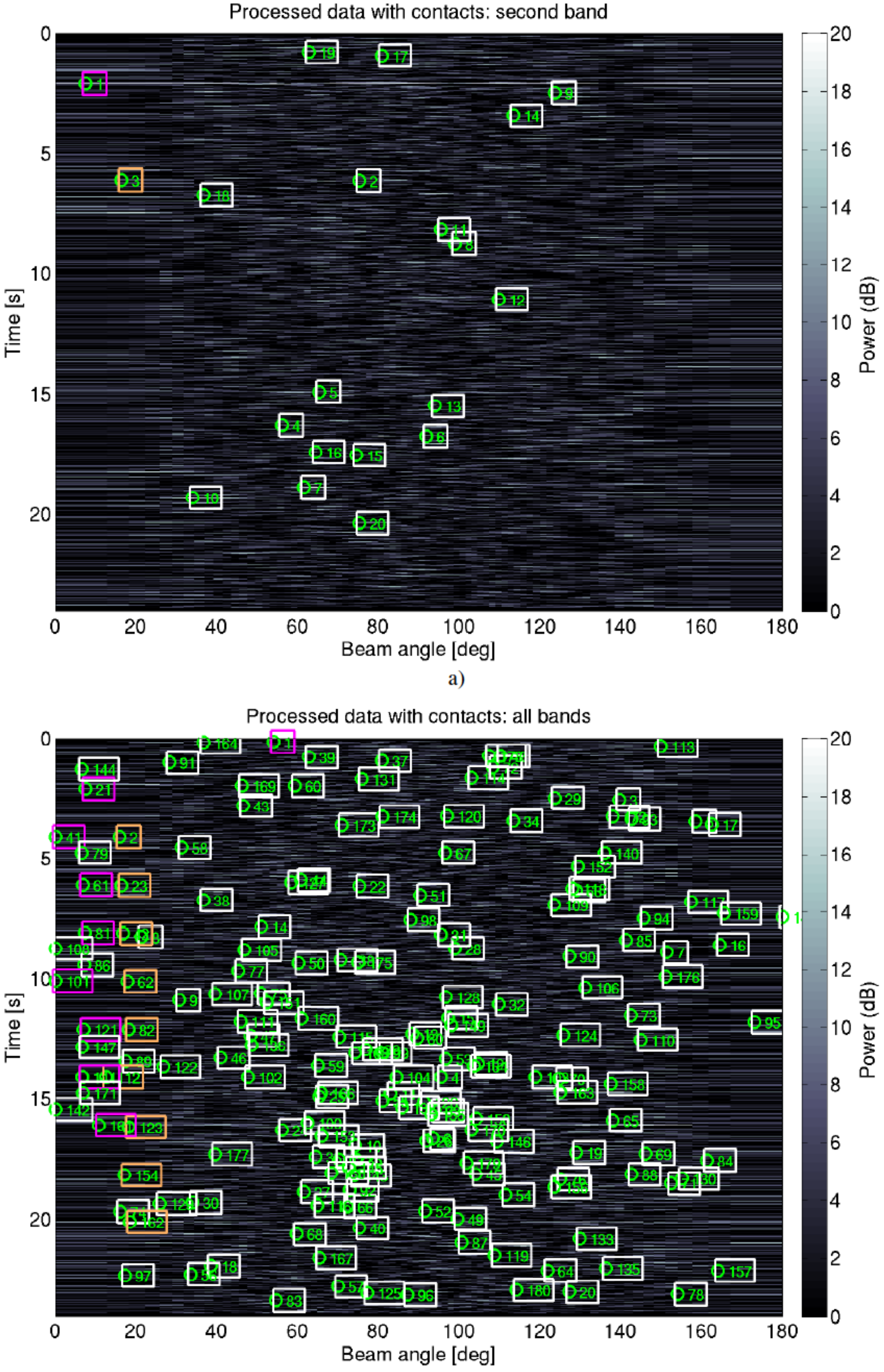


Fig. 3. a) Beamformed, matched-filtered, normalized data for a single sub-band (1878–1956 Hz) of the raw data shown in Fig 2. The 20 highest SNR contacts formed by `CAS-CAINPro` are overlaid (numbered in decreasing order of SNR). b) Beamformed, matched-filtered, normalized data for the ninth sub-band (2422–2500 Hz) with 180 contacts formed from all of the sub-bands overlaid.

TABLE I. THE TIME TAKEN BY THE CAINPro AND CAS−CAINPro LIBRARIES TO PROCESS A 24 s ATLAS FILE CONTAINING DATA FROM 64 ELEMENTS SAMPLED AT 10000.84 Hz. SUB-BAND PROCESSING WAS USED WITH NINE SUB-BANDS. VALUES WITH AN ASTERISK WERE ESTIMATED BY MULTIPLYING THE RUN TIME FOR A SINGLE SUB-BAND BY NINE. THE PC PROCESSOR IS AN INTEL® XEON® E5-2620 2.10 GHz WITH TWO CORES, AND THE AUV ON-BOARD PROCESSOR IS AN AMD QUAD-CORE 2 GHz PROCESSOR.

Process	CAINPro run time on PC (s)	CAS−CAINPro run time on PC (s)	 run time on AUV (s)
Initialization	11.70*	1.95	4.42
Data file conversion and detrending	0.77	0.49	0.66
TD basebanding / FFT + FD basebanding	3.46*	0.39	0.67
FD beamforming	2.03*	0.31	1.01
FD matched filtering + IFFT	0.15*	0.17	0.90*
Downsampling + normalization + clustering	0.28*	0.84	0.99*
Complete processing (including initialization/storing)	18.39*	4.65	7.98
Real-time processing	6.69*	2.20	4.23

*estimated

mid-frequency source, and echo repeater were all towed from NRV Alliance. The bearing estimates for the direct and echo repeater arrivals are relatively close to 0° (Approximately 10° and 20°) and the difference are caused by differences in tow position between the sources, the echo repeater and array.

The previous analysis of the first sub-band was extended to all nine sub-bands. Figure 3.b shows all 180 contacts obtained for the nine sub-bands. The data plotted in the background are the beamformed, matched-filtered, normalized data for the ninth sub-band, noting that the cluster reordering has not been performed on these contacts. The echo repeater contacts consistently occur every 2 s at a bearing of approximately 20° for all of the sub-bands. The direct arrival is also relatively consistent, except for the first sub-band where the bearing estimate is approximately 60°. This is an edge effect of the normalizer, whose window zeroes the first samples of the data, which contains the direct blast of the first sub-band. The results in Figure 3 confirm that contact formation by `CAS−CAINPro` is working as well as `CAINPro`.

The results presented in this section demonstrate that the `CAS−CAINPro` CAS processing algorithm detected the echo repeater, forming contacts at the proper time delay and consistent in bearing. This proves the software is functioning properly, and can perform in real-time as confirmed by the run-time measurements taken.

IV. CONCLUSIONS

This report described the `CAS−CAINPro` CAS signal processing suite developed to run in real-time on board the OEX AUVs. The new CAS processor was based on the `CAINPro` algorithm developed and thoroughly tested for PAS systems at CMRE. The modifications and optimizations to the PAS software, required to process a CAS signal in real time, were presented. The CAS software was tested for the first time during the REP14 Atlantic sea trial, where it successfully ran on board the OEX AUVs as well as on desktop computers on NRV Alliance. The CAS processor was demonstrated to detect a target (echo repeater towed by NRV Alliance), which it passed as contacts to CMRE's tracker. All computations can be completed safely within pulse repetition interval as required for real-time processing.

ACKNOWLEDGEMENTS

This work has been completed under the Cooperative Anti-Submarine Warfare (CASW) Programme, Project 1.

The authors would like to thank everyone involved in the REP14 Atlantic sea trial, including Vittorio Grandi (Engineering Coordinator), Gerardo Parise (Echo Repeater software), Ryan Goldhahn (REP14 Atlantic Scientist In Charge), Kevin LePage (CASW Project Leader), everyone in the Engineering Department, and the crew and captain of NRV Alliance.

REFERENCES

[1] P. C. Hines, "Experimental comparison of continuous active and pulsed active sonars in littoral waters," in *1st International Conference and Exhibition on Underwater Acoustics*, May 2013, pp. 51–58.

[2] M. J. Hamilton, S. Kemna, and D. T. Hughes, "Antisubmarine warfare applications for autonomous underwater vehicles: The GLINT09 field trial results," *Journal of Field Robotics*, vol. 27, pp. 890–902, 2010.

[3] M. Dukhan, "Yeppp! library documentation for C programmers," 2013, available at `http://www.yeppp.info/c/`.

[4] M. Frigo and S. G. Johnson, "The design and implementation of FFTW3," *Proc. IEEE*, vol. 93(2), pp. 216–231, 2005, available at `http://http://www.fftw.org`.

[5] M. R. Benjamin, P. M. Newman, H. Schmidt, and J. J. Leonard, "A Tour of MOOS-IvP Autonomy Software Modules," MIT Computer Science and Artificial Intelligence Lab, Tech. Rep. MIT-CSAIL-TR-2009-006, January 2009.

[6] T. E. Schneider and H. Schmidt, "MOOS-IvP communications software with acoustic networking from the goby underwater autonomy project," Technical Memorandum LAMSS-09-03, August 2010.

Document Data Sheet

Security Classification	*Project No.*

Document Serial No. CMRE-PR-2019-127	*Date of Issue* June 2019	*Total Pages* 6 pp.

Author(s)

Gaetano Canepa, Andrea Munafò, Stefan Murphy, Michele Micheli, Luca Morlando

Title

Real-time continuous active sonar processing

Abstract

This work describes the development of continuous active sonar (CAS) processing at CMRE. The software uses subband processing to achieve a faster update rate than is possible with pulsed active sonar (PAS). The software development was based on CMRE's PAS processing software, CAINPro, which has been thoroughly tested during previous sea trials and in postprocessing data analysis. Computational efficiency was carefully considered and many optimizations were made so that the software can run in real-time using the constrained computing resources on board CMRE's Ocean Explorer autonomous underwater vehicles (AUV). The software was successfully tested during the REP14 Atlantic sea trial in July 2014, and was able to demonstrate real-time detection of an echo repeater on all nine sub-bands that were processed. The CAS algorithm runs in real-time on the processing board installed on the AUV.

Keywords

Software, sonar, arrays, real-time systems, signal processing algorithms, frequency-domain analysis, repeaters

Issuing Organization

NATO Science and Technology Organization
Centre for Maritime Research and Experimentation

Viale San Bartolomeo 400, 19126 La Spezia, Italy

[From N. America:
STO CMRE
Unit 31318, Box 19, APO AE 09613-1318]

Tel: +39 0187 527 361
Fax:+39 0187 527 700

E-mail: library@cmre.nato.int