Real Time Time-Frequency Active Sonar Processing: A Massively Parallel Approach

George P. Zvara
Naval Undersea Warfare Center, Division Newport
Detachment New London
New London Ct. 06320

Abstract

A paradigm for massively parallel processing of matched filters, replica correlators, ambiguity functions and time-frequency distributions using an SIMD (Single Instruction Multiple Data) programming methodology is presented.

It is shown that active sonar detection algorithms, as implemented by frequency domain processing, can be a "natural match" to an SIMD methodology. Using the Connection Machine, an 8192 node SIMD parallel processor, this methodology holds great promise in meeting the extensive computational needs of enhanced active sonar systems.

The decomposition process is presented and examples, output of the computer program CMASP (Connection Machine Ambiguity Surface Processor), are given. CMASP can provide real time simultaneous multiple beam, Doppler and waveform replica correlations. Synthetic data are processed and the corresponding CMASP outputs are displayed as three-dimensional ambiguity surfaces on networked graphic workstations.

Because of efficient problem decomposition, in addition to the target bearing, range and velocity information as provided by continuous ambiguity surfaces, other time-frequency processing can be exploited. Specifically, instantaneous-like time-frequency distributions can be realized (e.g. Wigner, Rihaczek distributions) in which the data set is presented and processed as time-varying spectral reresentations.

1. Introduction

Sonar system engineers are anticipating an explosion of computational requirements for future sonar systems. Driven by multiple arrays or by advanced signal processing techniques (e.g. channel adaptive processing) future sonar systems will require real-time multi-tereflop capability. Evolving computer technologies, especially parallel processing regimes, may meet these requirements. However, these technologies must be evaluated as the sonar systems are being designed. Maximum performance may not be realized by a "backfit" solution. As sonar systems and computers co-evolve, they must be co-evaluated to ensure an optimal and efficient match.

Conventional serial processors are expected to be restricted in their capability to meet these future computational requirements. A single CPU (Central Processing Unit) based machine may be bounded by performance or by a physical (footprint) restriction. Parallel processing has become a viable option. However, a parallel processor is not "just another computer"; it is representative of a separate and distinct technology. The system designer will have to choose among several "types" of parallel architectures. Some will provide an efficient solution, others will not. Machine dependent software, representing a parallel mindset, will have to be evaluated. Algorithm decomposition methodologies will have to be evaluated as well. Some design inquiries will be:

- What type of parallel processor is "best" for the application? Is there a "natural fit"?
- Is the machine easy to program? What kind of tools are there to help map the application onto the multiple arithmetic processors?
- What is affordable from a time, money, footprint perspective ?

Therefore, the system designer, as part of the design process, will have to research the computer options to achieve an acceptable match of the computational requirements. This paper serves to introduce the reader to a type of parallel programming methodology, Single Instruction Multiple Data (SIMD) as implemented on a massively parallel computer (Connection Machine). The report presents examples describing the decomposition of signal processing requirements to achieve effective use of parallel computer architectures.

1.1 Parallel Computer Architectures and Taxonomy

A full discussion of parallel signal processing is beyond the scope of the paper, however the reader is directed to Flynn's [1] classification of parallel processing systems. Discussion is restricted to parallel processing systems; defined here as processing that is partitioned among multiple processing units (P.U.) that are within a single computer. Distributed processing is not considered. Distributed processing is defined here as computations that are partitioned among multiple standalone computers. A viable option for implementing sonar signal processing functions, the use of distributed processing is, however, a system level design issue.

Flynn classifies parallel processing architectures into four organizational classes. Experience suggests a fifth class (Figure 1). As will be seen, the classes are defined by the relationship of data and instruction types within an architecture of multiple PUs. The classes are:

- *SISD* - Single Instruction Single Data: A parallel processing architecture type such that at a given instant of time each of N PUs, within the architecture, is executing the same instruction (Single Instruction). The instruction uses data that is common across all PUs. Conceivably one would use this type of architecture in mission critical systems insuring reliability with fault tolerant redundant computing.

- *SIMD* - Single Instruction, Multiple Data: This architecture contains configured PUs that execute, in a lock step, the same instruction on data that is different across the PUs. The Connection Machine (CM), developed by Thinking Machines Inc., is an example of this type of architecture. The architecture can be subclassified as fine-grained. Because an individual PU may be limited by, for example, memory or word size, the power of the machine comes from thousands of massively parallel processors simultaneously solving the problem the same way. A weakness may be in problem decomposition, namely, the application may or may not be optimally decomposed by a SIMD approach. However, with the singular instruction restriction, the state of the machine (e.g. executing instruction) at any instant of time is known. Such a feature is desirable, guaranteeing the state predictability at arbitrary test points of the system.

- *MISD* - Multiple Instruction Single Data: A hypothetical class of architectures such that autonomous PUs are executing dissimilar code on the same data. To date it has not been realized.

- *MIMD* - Multiple Instruction Multiple Data: A class of architecture such that autonomous PUs are asynchronously executing dissimilar code on dissimilar data. MIMD machines have the potential to be extremely powerful and can be highly competitive with SIMD architectures. MIMD machines have multiple (tens to hundreds) of high performance parallel PU engines. The PUs are usually configured in a source-sink arrangement. A disadvantage for these machines is the difficulty of efficient algorithmic decomposition across the PUs. As most MIMD machines have processors that act as sources and sinks for each other, there is a potential for bottlenecks to occur. If programmed inefficiently, MIMD machines can become "unbalanced" as some downstream processors wait for the result from an overtaxed upstream PU. Optimizing or balancing the burden across multiple PUs for maximum performance may be an expensive and difficult process.

- *SPMD* - Single Program Multiple Data: This is a new class of programming methodology resulting from experience with programming MIMD architectures. Used with MIMD machines, the SPMD methodology allows for multiple copies of the same instruction set (computer code) to run asynchronously on PUs having different input data sets. SPMD programming can be viewed as a compromise between SIMD and MIMD methodologies. It allows for an easier decomposition process for programming complex MIMD architectures, for example, by assigning data channel processing flows on multiple unlinked asynchronous PUs.

1.2 Active Sonar As An Application

It is well known that the optimal receiver for active sonars operating against gaussian noise is the matched filter [2]. A replica of the transmitted waveform is bandpassed matched filtered, by way of replica correlation, with the received signal. A square-law envelope detector is then applied. If the result is above a user defined threshold, a detection is called. Similar processing is done for each desired spatial (beam) direction.

For each beam, the processing can be extended to a two dimensional problem of time and frequency. The time dimension measures the delay of the transmitted waveform in the received signal. This delay is proportional the range to the target. Time elongation ("stretching") in the observed signal may be proportional to extended targets. The spectra of these targets will show selective frequency fading. Such targets are commonly referred to as range-spread or dispersive targets.

The frequency dimension measures how a observed signal differs in spectral content from the transmitted signal. The returned signal can contain a shift in frequency (up or down), the Doppler effect, that is proportional to the relative motion between target and sensing platform. Further, if the orientation of the target changes as a function of time, the returned signal is amplitude modulated. The spectrum of such a return is spread in frequency. The amount of spreading is proportional to the rate of change of a target's reflective characteristics. Such a target is commonly called a Doppler-spread target.

For an active sonar system, it is common to transmit sequential waveforms types, each type being sensitive to the range or the motion parameter. For example, a CW waveform, sensitive to Doppler motion, can be transmitted and followed by a HFM (hyperbolic frequency modulated) waveform. The HFM is sensitive to range but relatively Doppler insensitive. Thus, an active sonar correlation receiver must be flexible and robust enough to correlate different waveform types.

The acoustic Channel environment can augment the transmitted waveform in a likewise manner. However, for purposes of discussion we will restrict the discussion to the effects of the target.

Time (τ)-frequency (ω) processing can be realized by the creation of a bank of matched filters and square-law envelope detectors. The single processing channel for replica r(t)

$$\Re_{rs}(\tau) = \int_t r(t)\, s^*(t-\tau)\, dt^{1}$$

is augmented by a bank of parallel matched filters that differ by the replica Doppler hypothesis value υ , $\omega = 2\pi\upsilon$. Then

$$\Re_{rs}(\tau, \omega) = \int_t r(t) \exp[-j\omega t]\, s^*(t-\tau)\, dt\,. \qquad (1)$$

For purposes that will be made clear later, the frequency shift operator is absorbed into the replica.

$$\Re_{rs}(\tau, \omega) = \int_t r_{\omega}(t)\, s^*(t-\tau)\, dt \qquad (2)$$

where

$\omega = 2\pi\upsilon,$

[1] All intergrals presented reflect intergration from -∞ to +∞; the notation beneath the intergral symbol assists in assessing which variable is being intergrated.

$r_{\omega}(t)$ = replica of the transmitted waveform at frequency ω
$s(t)$ = received signal,
$\Re_{rs}(\tau, \omega)$ = the two dimensional output of the matched filter for frequency ω at time t sometimes referred to as the matched filter uncertainty function (MFUF).[3].

It is common for the magnitude square of $\Re_{rs}(\tau, \omega)$ be plotted as a three dimensional surface of time, frequency vs correlation value. Such a plot is representative of the matched filter ambiguity function (MFAF). The MFAF can be viewed as the output of a range envelope detector for multiple frequency shifted correlators.

A distinction is made between the symmetric and asymmetric versions of the uncertainty function and between the auto and cross correlators, i.e. between the auto and cross ambiguity functions. These distinctions are often obfuscated in the literature. Equation 1 represents the delay asymmetric cross correlation of a received signal s(t) with a Doppler shifted replica r(t), the kernel processing in the context of active sonar systems. However, many researchers in dealing with "generic" processing, define the uncertainty (and therefore the ambiguity) functions by using the following symmetric autocorrelation notation.

$$\Re'_{ss}(\tau, \omega) = \int_t s(t+\tau/2)\ \ s^*(t-\tau/2)\ \exp[-j\omega t]\ dt\ . \qquad (3)$$

Equation 3, auto correlates the unknown signal s(t) and is symmetric about the delay parameter τ. The notation $\Re'$ is used for symmetric delay notation.

For purposes of later discussion of the Wigner distribution, (3) is modified for the cross correlation case for a known signal r(t)

$$\Re'_{rs}(\tau, \omega) = \int_t r(t+\tau/2)\ \ s^*(t-\tau/2)\ \exp[-j\omega t]\ dt\ . \qquad (3a)$$

The distinction is made for two reasons. First, (1) is more appropriate for an active sonar time-Doppler matched filter and is simpler to implement (3a). Equation 3 is better used for passive sonar (unknown signal) applications. Secondly, as will be shown, instantaneous time-frequency processing for symmetric correlations must implement the Wigner operations; asymmetric correlations must implement the Rihaczek operations. It will be shown that the results of these operations will be similar but phase shifted in the instantaneous time-frequency domain.

2. Method: Fast Correlation as a Natural SIMD Process

Equation 2 is written so that the frequency shift operator is absorbed in the replica.

correlation:

$$\Re(t,\omega) <-> \mathbf{R}(f,\omega) = R_{\omega}(f)\ S^{*}(f) \tag{4}$$

where

$\mathbf{R}(f,\omega) <-> \Re(t,\omega)$
$R_{\omega}(f) <-> r_{\omega}(t)$ and
$S(f) <-> s(t)$ are one dimensional Fourier Transform pairs with
$<->$ the Fourier Transform operator.

With the advent of digital serial (von Neumann) computing in concert with the highly efficient Cooley-Tukey radix two Fast Fourier Transform, fast correlation has became an appealing substitute for conventional time based correlation methods. However, computational performance by serial (von Neumann) computer architectures is still limited. Increases in performance have been directly tied to faster PUs and this enabling technology is peaking. Serial architectures (almost by definition) demand that frequency domain processing be tied to a large "DO LOOP" (of size N for DFT size N) where, after Fourier transformation, the computer program continues to cycle through f_i for i=0,1,2 ... N-1. The processing is bound by architecture and cannot be further partitioned.

Fast correlation, shown by (4), decomposes into a two operator process. For and at every temporal frequency f:

1. Conjugate each transformed data for example S(f): $S(f)->S^{*}(f)$

2. Complex multiply: $R_{\omega}(f)\ S^{*}(f)$.

The process distills to three independent (non-recursive, i.e. data independent) multiply operations. In a timeline sequence of computer assignments each step can be thought as a singular individual machine instruction. The instructions are performed on the data as represented by $R_{\omega}(f) \times S^{*}(f)$.

Inspection of this decomposition with that of the five classes of parallel computer architectures suggest that a complex multiply is optimal as an SIMD process. A one-to-one mapping of the temporal and Doppler spectral component f_{ω} to processor $p_{f,\omega}$ could be assigned. That is, the processor $p_{f,\omega}$ could be "responsible" for the processing of temporal frequency f at Doppler frequency ω. Such a partitioning of data to processors reflects the "multiple data" part of the SIMD class name. The "single instruction" aspect is reflected by the single operation (multiply) that is simultaneously performed in lock step by all processors. Thus N effective multiplies are performed at the cost of one instruction. Compared to conventional serial architectures, processing time is "amortized", in some sense, by N processors.

Frequency domain multiplication is the basis for other important signal processing problems including:

- convolution $\mathfrak{Z}(t)$ <-> $X(f)\ Y(f)$
- phase shift beamforming (time delay equivalent):

  $x_i(t-\tau_i)$ <-> $X_i(f)\ \exp[-j2\pi f\tau_i]$ for delay τ_i at hydrophone i
- generic band filtering: $\eta(t)$ <-> $X(f)\ H(f)$
- power calculations: $Z(f) = X(f)\ X^*(f)$

Here x(t), y(t) are arbitrary time series with Fourier Transform pairs X(f), and Y(f). H(f) is the transfer function of a lowpass, band pass filter.

To fully exploit SIMD machines, the signal processing kernels must be distilled to an intrinsic SIMD methodology, as shown in the above examples. Additionally, there is further partitioning of parallel data channels (for example multiple beams, hydrophone elements etc.) to multiple PUs. Examples are shown in the next section.

2.1 Connection Machine Architecture

The Naval Undersea Warfare Center's high performance computing resources include a Connection Machine (CM) model CM200. Hosted by a Sun4 workstation, the NUWC CM configuration consists of 8192 physical processors with 1M bits of local memory per processor. The word length is normally 32 bits per word, optionally to 64 bits per word. The programming languages for the CM are C* (pronounced "C-star", a parallel version of C), and parallel FORTRAN or LISP. Within a program the C/C* language demarcation is clear: at runtime the parts of the program that run on the SUN host (e.g. data preparation) are defined by C code. The part which runs in the CM (including "reading from" and "writing to" the host) is written in C*. The host interfaces with mass storage devices that hold real or synthetic echo data. Additionally, the host is networked to graphic workstations (Silicon Graphics) which will display in real time the processed data.

A key feature of the CM is virtual processing capability. Virtual processor capability is the ability of a physical processor to act as if it were N virtual processors (VP). In the Connection Machine system, the user writes code that defines the "Virtual Processor Ratio" (VPR) - the ratio of virtual processors to physical processors. The use of VP is essential for optimal SIMD processing. For example, in Figure 2 if a matrix of 1024 frequencies by 64 hydrophone channels is mapped one-to-one to the required processors, 64K processors are needed. Thus the VPR equals 8 (64K/8K) and each physical processor acts as if it were 8 virtual processors. The memory available to each VP is reduced by a factor of 8 from that of the physical processor: 1 M bit per physical processor / 8 = 128K bits per virtual processor. Only the first 8192 virtual processors act in a SIMD fashion; the remaining virtual processors operate in SIMD fashion at the next clock cycle in batches of 8K. However, this execution fragmentation is transparent to the user and for all effects all 64K virtual processors act as a single SIMD computer.

To fully exploit this virtual processor capability, the CM allows the N virtual processors to be "organized" into virtual geometries or shapes. The simplest shape is a 1-dimensional organization of virtual processors, a row vector of shape (1 x N). Communication (i.e. data transfer) between virtual processors of a given shape is supported and if need be, processors can send results of an operation to nearest or extended neighbors.

A two-dimensional grid virtual geometry is the next higher order of shape; for example, M virtual processors allocated for the rows (y-axis) and N for the columns (x-axis), Figure 2. The total number of virtual processors are then N x M. A clear use of this shape would be to merge parallel signal processing parameters (e.g., frequency) with parallel spatial or temporal parameters (e.g channels of data or beams). For example, a virtual processor at position p(x,y) is assigned for processing frequency x of hydrophone channel y.

There is a Connection Machine restriction as to the choice of axis length for any geometry. Any axis length (corresponding to the total number of processors in that direction) must be a power of two.

An example of the use of a three-dimensional shape is the processing of a phase shift beamformer, Figure 3 . Processors partitioned along the three axes represent frequency, hydrophone channel and beam steer direction parameters [4]. The amount of phase shift (time delay) for a given beam steer direction is a function of frequency, f_i, and hydrophone position, h_k. Each processor could calculate the appropriate phase shift for its position in the virtual processor grid and then apply these shifts to the incoming (frequency domain) data. These steering weights need only to be calculated once and are held constant during runtime. An inverse Fourier transform across the frequency axis is then applied. This delayed time series is then summed across the hydrophone axis. The result are beamformed data with each beam pointing to the desired position in beamspace.

The Connection Machine has the capability to configure processors in shapes higher than three dimensions; a four dimensional shape is detailed next. A 31-dimensional geometry is the maximum shape the CM will support.

2.2 Connection Machine Ambiguity Surface Processor - CMASP

This section describes the use of the Connection Machine as a multiple beam, multiple waveform time-frequency correlation receiver as realized by the computer program CMASP: Connection Machine Ambiguity Surface Processor.

CMASP processing assigns to a single Connection Machine virtual processor a particular instance of the independent variables of a multiple beam, multiple doppler, multiple waveform replica correlation. The independent variables are time/temporal frequency, Doppler frequency, waveform type and beams. N Connection Machine virtual processors are partitioned into a four-dimensional shape whose axes are defined by the aforementioned parameters. N is defined by the product of the number of time (T), doppler frequency (D), waveform (W) and beam (B) cells.

The kernel processing for CMASP is matched filtering by way of fast correlation. Previously, it was shown that fast correlation has a natural SIMD structure. A given Doppler shifted replica and received signal are transformed into the frequency domain about temporal frequency f. The complex representations of the replica and received signal are the multiple data components of the SIMD methodology; the conjugate complex multiply between the two is the single instruction executed by each virtual processor for all frequencies f.

A key component that optimizes the SIMD and virtual processor architecture is the two-dimensional parameter set representing the replicas. For a given waveform, the replicas are identified and bounded by the time and Doppler frequency axes. This information is labeled the replica template. For an arbitrary waveform, the template is a suite of time series replicas each shifted by a Doppler hypotheses. These templates are the reference portion of the replica correlation and they are assigned to specific virtual processors within the four-dimensional shape. Their values are calculated at the start and are fixed during processing; there is no need to update them during runtime. However, if new waveforms are transmitted in the next ping cycle, these replica templates must be updated accordingly.

For a given transmitted waveform, the replica template is replica correlated in parallel across all beams figure 4. This is done for every waveform. The final result, computed in parallel, is multiple beam, multiple Doppler, multiple waveform replica correlation based on the kernel SIMD structure of fast correlation. The output is formatted and collected so the result is displayed as multiple ambiguity surfaces representations for multiple waveforms across multiple beams.

3. Resuslts: CMASP Processing Size and Timing

Experience shows, that, for CM processing, the size of a problem that can be considered is proportional to the amount of memory available in the machine. Certain applications, especially ones with a "natural" SIMD "fit", are more memory bound than compute bound. Stated differently, the size of an application that can be processed is determined by the amount of total memory available, not by the computational resources. Computational resources tend to determine the real time performance aspects of a given problem.

The following is the upper size limit for this decomposition of replica correlation on a CM200

Number of Waveforms (Codes), W	4
Number of Beams, B	8
Number of Doppler Channels, D	16
Number of Time/Frequency Cells, F	8192

Thus N equals 2^22[2] (4 x 8 x 16 x 8192) and represents the number of virtual processors that must be created at runtime. The virtual processor ratio (VPR) here equals 512 (2^22 - 2^13). The VPR is important as it defines how much memory each virtual processor is alloted from the total memory of each physical processor. For a CM200, each physical processor has 2^20 bits of memory. Governed by the VPR ratio, for this case 512, each virtual processor is alloted 2048 bits (2^20 -2^9 = 2^11). At 32 bits per word, each virtual processor has enough memory for 64 words. CMASP has been optimized to use less than 64 words per virtual processor.

The CM architecture limits the parameter increase to the next power of two. For example, if one wished to increased the number of beams from the current 8 to arbitrarily 10, 16 beams worth of memory must then be reserved. All else being equal, the VPR then would double to 1024, halving the amount of local memory for each virtual processor to 1024 bits or thirty-two 32-bit words per virtual processor. Currently CMASP processing has not been optimized for 32 words per virtual processor Thus the process becomes memory bound for the 10-beam case.

However the 10-beam case can be processed if relief is provided to the size of any of the other parameters, for example, by reducing the number of time/frequency bins to 4096. Doubling the number of beams to 16 while halving the time/frequency bins maintains the VPR at 512 with likewise memory allotment and capacity. Thus for a given coding efficiency the size of a problem is not governed by the lengths of individual parameters as much as the product of all parameter lengths, here N.

3.1 CMASP Display Presentation

The output of CMASP processing are matched filter ambiguity functions presentations across multiple beams for sequential transmitted waveforms within a ping . The CM host is networked to graphic display workstations running custom designed display code. The CM host and graphic workstations run in a client-server relationship that is common in UNIX socket communications.

Figure 5 assists in the description of the CMASP displays. The ambiguity field for each beam is fanned out horizontally across the screen; time (waterfall) is along the vertical axis with the oldest data along the lower part of the display. The user selected A-scan window is at the far right in alignment with waterfall time. Within each beam display 16 "stripes" are shown corresponding to different Doppler channels. Zero Doppler is represented by the middle stripe; closing Doppler is to the left of the middle stripe, opening to the right. Each pixel is color coded proportional to the magnitude of the correlation value at that time. (The

[2] The notation A^B is used to describe exponentiation: A to the power of B.

quantization level map is not shown in Figure 5 but is presented on the actual displays.)

Captured screen representation of CMASP, output for the three processed waveforms, can are presented in Figures 6 through 8. It is underscored that the data is continuously updated and calculations are performed in parallel; the data is displayed concurrently as available.

Some minor notes on the display complete the presentation to the viewer. Note the quantization map at the upper limb of the screen which maps color to correlation value. Above the A-scan is the user selected beam and Doppler channel selection numbers. On the far left two numbers in a ratio format are shown (e.g. 12/17). This identifies the block number displayed with respect to the total number of blocks available to be processed . A block represents approximately 2.0 seconds of processed data. Processed blocks are separated by the small white horizontal lines, left of Beam 0 and right of Beam 7. Finally, red arrows point to which time segment has been centered for a "zoomed Doppler view" which is displayed at the lower edge of the screen. Across all beams and Doppler, five time pixels above and below the selected time are expanded for a "zoomed view" to assist in Doppler interpretation. It should be noted that the computing is not zoomed.

3.2 Display Example

Figures 6, 7 and 8, screen captures from the display devices, present processed CMASP results for synthetic data. A signal generator injected the return from two point sources in a normally distributed Gaussian background. The ping consisted of abutting waveforms: a CW, a HFM and a 12-Chip linear congruence frequency hop code. A 3 dB target was centered at beam 2 at a time of 10.5 seconds with 4 knots closing motion. A 0 dB target was centered at beam 5 at 15 seconds. There is an arbitrary 3 dB roll-off between adjacent beams.

Figure 6 shows the return for the HFM waveform. The slanting return across the Doppler channels reflects the Doppler-range coupling of HFM waveforms. Note this waveform's Doppler insensitivity but its strong range discrimination. The A-scan at the right shows the deflection for beam No. 2 and Doppler channel 5 (+4 knots). The fainter target can be seen at the upper right centered in beam 5. The numerous yellow horizontal lines are clutter returns across Doppler and beams. A Doppler zoom view is displayed below each beam return. The red arrows indicate the region that is highlighted.

Figure 7 shows the multi-beam, time-frequency processing for the frequency hop code (FHC). Immediately abutting the HFM, the FHC is an experimental 12-chip linear congruence code. This experimental waveform is used in an attempt to discriminate for both range and motion. An expected detection occurs along the time parameter. Discrimination across Doppler is not quite as clear; the waveform cannot completely disambiguate relative motion.

Figure 8 presents the processed return for the leading CW waveform of the ping.

The waveform correctly detects the motion of the hypothetical targets in Doppler channel No. 5 of beam Nos. 2 and 5. As expected, the CW is a poor discriminator for range as shown by the broad triangular correlation.

It is underscored that these results are processed simultaneously and updated in real time. A fourth redundant waveform is also processed but the results, due to laboratory hardware constraints, is not presented on a graphic terminal.

4. Discussion: Enhanced CMASP Processing: Instantaneous-like Time-Frequency Distributions

After achieving real-time access to uncertainty functions a powerful yet simple extension can be realized.

Instantaneous-like time-frequency distributions have applications for novel detections and classification schemes. Analogous to a musical score, time-frequency signal distributions present the nature of signals over a time-frequency plane. The combined time domain and frequency domain analyses yield a revealing picture of a signal's temporal and spectral components [5].

Various operators are available to transform the data to the time-frequency domain. A complete discussion of this the subject is beyond the scope of this paper; the reader is encouraged to consult Hlawatsch for a fine tutorial on the subject. However, we will refer to two time-frequency transformation methods the: Rihaczek distribution and the Wigner distribution [5,6].

The Rihaczek distribution, also known as the complex-energy density function, is defined as:

$$\varepsilon\ (t,f) \equiv \int_{\tau} x(t)\ x^*(t-\tau)\ \exp[-j2\pi\tau f]\ d\tau \qquad (5)$$

which can be thought as the two-dimensional time-frequency auto convolution of x(t) [Rihaczek].

For our discussion of a two-dimensional replica correlator, we extend (5):

$$\varepsilon_{r,s}\ (t,f) \equiv \int_{\tau} r(t)\ s^*(t-\tau)\ \exp[-j2\pi\tau f]\ d\tau \qquad (6)$$

where

r(t) is the replica for a given transmitted waveform,
s(t) is the received signal in a single beam,
$\varepsilon_{r,s}$ (t,f) is the time-frequency Rihaczek distribution
τ is the time delay factor and,

f is the frequency shift.

Now, rewriting (1) with $\omega = 2\pi\upsilon$ and using the subscripts r and s for replica and signal;

$$\Re_{r,s}(\tau, \omega) = \int_t r(t)\, s^*(t-\tau)\, \exp[-j2\pi\upsilon t]\, dt$$

using the Dirac delta function $\delta(t)$ and its integral property

$$\Re_{r,s}(\tau, \omega) = \int_t \int_\tau r(t)\;\; s^*(t-\tau)\, \exp[-j2\pi\upsilon t]\; \delta(\tau_1 - \tau)\, dt\, d\tau$$

now substituting the delta function with its Fourier integral representation

$$\Re_{r,s}(\tau, \omega) = \int_t \int_\tau \int_f r(t)\;\; s^*(t-\tau)\, \exp[-j2\pi\upsilon t]\;\; \exp[-j2\pi(\tau_1 f - \tau f)\,]\, dt\, d\tau\, df$$

Finally using (6)

$$\Re_{r,s}(\tau, \omega) = \int_t \int_f \varepsilon_{r,s}\,(t,f)\, \exp[-j2\pi(\upsilon t - \tau f)]\, dt\, df \qquad (7).$$

Equation 7 shows that the Rihaczek cross complex energy density function, $\varepsilon_{r,s}(t,f)$ and the matched filter uncertainty function, $\Re_{r,s}(\tau, \omega\ )$ are double Fourier Transform pairs. Thus the time-frequency distribution of the two-dimensional replica correlator can be retrieved from the matched filter uncertainty function by transforming forward from delay τ to frequency f and inversely from Doppler shift υ to time t.

Alternatively, the two-dimension time-frequency distribution of the two-dimensional correlator can be written as a SIMD operation:

Using (6),

$$\varepsilon_{r,s}\,(t,f) = \int_\tau r(t)\, s^*(t-\tau)\, \exp[-j2\pi\tau f]\, d\tau$$

$$= r(t) \int_\tau s^*(t-\tau)\, \exp[-j2\pi\tau f]\, d\tau$$

$$= r(t) \int_\tau s^*(\tau)\, \exp[-j2\pi\tau f]\, \exp[-j2\pi ft]\, d\tau$$

$$= r(t)\, S^*(f)\, \exp[-j2\pi ft] \qquad (8)$$

For implementation purposes, the replica time series, r(t), and the phase shift template, exp[-j2πft] are constants and can be precalculated and stored in appropriate processors. With

$$\xi(t,f) \equiv r(t) \exp[-j2\pi ft]$$

(8) becomes

$$\varepsilon_{r,s}(t,f) = \xi(t,f)\, S^*(f)\ .$$

Thus a dedicated time-frequency SIMD processor needs only to multiply the conjugated spectrum of the received signal with the constant template $\xi(t,f)$.

Analogously, it can be shown that the Wigner distribution is the double Fourier Transform pair with the symmetric form of the two-dimensional correlation function. Using the two-dimensional symmetric cross correlation uncertainty function (3a)

$$\Re'_{r,s}(\tau, \upsilon) = \int_t r(t + \tau/2)\, s^*(t - \tau/2) \exp[-j2\pi\upsilon t]\, dt \tag{9}$$

The cross Wigner distribution can be shown to be

$$W_{r,s}(t,f) = \int_t \int_f \Re'_{r,s}(\tau, \upsilon) \exp[j2\pi(\upsilon t- \tau f)]\, dt\, df \tag{10}$$

or the Wigner distribution, $W_{r,s}(t,f)$, is the double Fourier transform of the symmetric uncertainty function.

The relationships between the Wigner and Rihaczek distributions and the symmetric and asymmetric versions of the uncertainty functions are discussed below.

Taking the symmetric version of the two dimensional uncertainty function (9)

$$\Re'_{r,s}(\tau, \upsilon) = \int_t r(t + \tau/2)\, s^*(t - \tau/2) \exp[-j2\pi\upsilon t]\, dt$$

letting $t_1 = t + \tau/2$

$$\Re'_{r,s}(\tau, \upsilon) = \int_t r(t_1)\, s^*(t_1 - \tau) \exp[-j2\pi\upsilon(t_1 - \tau/2)]\, dt_1$$

substituting for (1) then

$$\Re'_{r,s}(\tau, \upsilon) = \Re_{r,s}(\tau, \upsilon) \exp[-j2\pi\ \tau/2] \tag{11}$$

i.e. the two-dimension symmetric cross correlation function $\mathfrak{R}'_{r,s}(\tau, \upsilon)$ is related to the asymmetric cross correlation function $\mathfrak{R}_{r,s}$ by the constant phase shift factors υ and τ.

Further, it can be shown that the Wigner and Rihaczek distributions have a similar relationship. Expressing the Rihaczek distribution in terms of the asymmetric uncertainty function, from (7),

$$\varepsilon_{r,s}(t,f) = \int_{\tau}\int_{\upsilon} \mathfrak{R}_{r,s}(\tau, \upsilon) \exp[j2\pi(\upsilon t - \tau f)]\, d\upsilon\, d\tau$$

using equation 10 and substituting for the asymmetric uncertainty function $\mathfrak{R}_{r,s}$,

$$\varepsilon_{r,s}(t,f) = \int_{\tau}\int_{\upsilon} \mathfrak{R}'_{r,s}(\tau, \upsilon) \exp[-j2\pi\upsilon\tau/2] \exp[j2\pi(\upsilon t - \tau f)]\, d\upsilon\, d\tau$$

from (10) in terms of the Wigner distribution of υ' and τ'

$$\varepsilon_{r,s}(t,f) = \int_{\tau}\int_{\upsilon}\int_{t}\int_{f} \mathbf{W}_{r,s}(t,f) \exp[-j2\pi(\upsilon' t - \tau' f)] \exp[-j2\pi\upsilon\tau/2] \exp[j2\pi(\upsilon t - \tau f)]\, d\upsilon\, d\tau\, dt\, df$$

collecting terms for the Dirac delta time and frequency functions,

$$\varepsilon_{r,s}(t,f) = \int_{\tau}\int_{\upsilon}\int_{t}\int_{f} \mathbf{W}_{r,s}(t,f) \exp[-j2\pi(\upsilon' t - \upsilon t)] \exp[-j2\pi\upsilon\tau/2] \exp[j2\pi(\tau' f - \tau f)]\, d\upsilon\, d\tau\, dt\, df$$

$$\varepsilon_{r,s}(t,f) = \int_{\tau}\int_{\upsilon} \mathbf{W}_{r,s}(t,f)\, \delta(\upsilon' - \upsilon)\, \delta(\tau' - \tau) \exp[-j2\pi\upsilon\tau/2]\, d\upsilon\, d\tau$$

$$\varepsilon_{r,s}(t,f) = \mathbf{W}_{r,s}(t,f) \exp[-j2\pi\upsilon\tau/2]$$

which is similar to the result of (11) in that the phase shifts between the symmetric and asymmetric uncertainty functions are maintained in their respective time-frequency distributions.

An example of the Rihaczek distribution calculated by the Connection Machine in real time is presented as a block waterfall in Figure 9. Three time histories are shown for two beams of a synthetic returned signal. The data set is identical to that processed for Figures 6,7,8. The plot shows the real time Rihaczek processing for two beams for three matched filtered waveforms: a CW, a HFM and a 12-chip FHC. For each block processing interval and for each code, the time axis (relative to that replica) is along the horizontal axis and the frequency axis is along the vertical. The plot shows the distribution across the time-frequency plane. The CW waveform (labeled Code 0) is represented as a strong single line constant across time, indicative of its monotone characteristic. The downswept HFM (Code 1) shows a sweep across time, though the resolution is not fine enough to show its hyperbolic nature. The experimental 12-chip FHC (Code 2) shows strong returns, discreet enough to ascertain the twelve chips and its linear congruence.

Note the changes in levels for these returns across time history and beam space, and that the peak returns for these codes are out of frequency band. This out of band structure reflects the out of band structure of the transmitted waveform.

4.1 Conclusions

Several signal processing operations including fast correlation and fast convolution have be presented in an SIMD methodology. A SIMD phase shift beamformer paradigm has been presented as well.

A multi-beam multi-waveform two-dimensional (time-frequency) correlation receiver has been programed using commercial off the shelf hardware, the Connection Machine. This receiver, called CMASP (Connection Machine Ambiguity Surface Processor), can process synthetic data and display, in real time, matched filter ambiguity functions. Providing range and Doppler motion coverage, these displays can also be used to explore the characteristics of Doppler-spread and range-spread targets for multiply transmitted waveforms.

A Single Instruction Multiple Data (SIMD) partitioning process for the CMASP correlator has been presented. The key to successful (optimal) use of an SIMD architecture is to decompose, in an SIMD fashion, the signal processing algorithm. The CMASP correlation receiver uses fast correlation (frequency domain) techniques as the decomposition process. Further, optimal processing is achieved by assigning a Connection Machine virtual processor to every independent variable in the problem set.

It has been shown that by using these techniques it is possible to process in real-time multi-beam, multi-waveform instantaneous time-frequency distributions. Specifically, the Rihaczek distribution has been presented. Instantaneous-like time-frequency presentations show promise for novel techniques for detection and classification of targets. These presentations can show the fine time-frequency response from known and unknown signals.

Evolving high performance computer architectures will enable real-time performance for computationally extensive conventional and advanced signal processing algorithms for the next generation sonar systems. That these devices are commercial off-the-shelf, reprogrammable and system embedable suggests that these systems will be affordable as well.

5. Acknowledgement

The author wishes to acknowledge D.H. Kim for assistance in programming CMASP and for writing the computer display code. Dr. Duncan Sheldon of the Naval Undersea Warfare Center and Dr. Edward Titlebaum of the University of Rochester are thanked for invaluable discussions and helpful suggestions.

This research was carried out in the Software Engineering Laboratory of the Submarine Sonar Department of the Naval Undersea Warfare Center, Detachment New London Ct., USA under the sponsorship of the US Office of Naval Research Technology Directorate, Associate for Undersea Surveillance.

6. References

[1] Flynn, M.J. "Very High-Speed Computing Systems", Proceedings of the IEEE, v. 54, No.12, December 1966.

[2] Van Trees H.L., Detection, Estimation and Modulation Theory Part III, John Wiley, New York, 1971.

[3] Cook and Bernfield, Radar Signals: An Introduction to Theory and Applications, Academic Press, New York, 1967 p.56.

[4] Zvara, G.P., Very Massively Parallel Acoustic Architectures (VMPAA): Candidate Applications, NUSC TM No 88-1145. Naval Underwater Systems Center, New London Ct. 06320. (UNCLASSIFIED)

[5] Hlawatsch F., Boudreaux-Bartels G.F., Linear and Quadratic Time- Frequency Signal Representations, IEEE Signal Processing Magazine, April 1992 p. 21-67..

[6] Rihaczek, A.W. Signal Energy Distributions in Time and Frequency, IEEE Transactions on Information Theory, V. IT-14, No. 3, p. 369-374 May 1968.

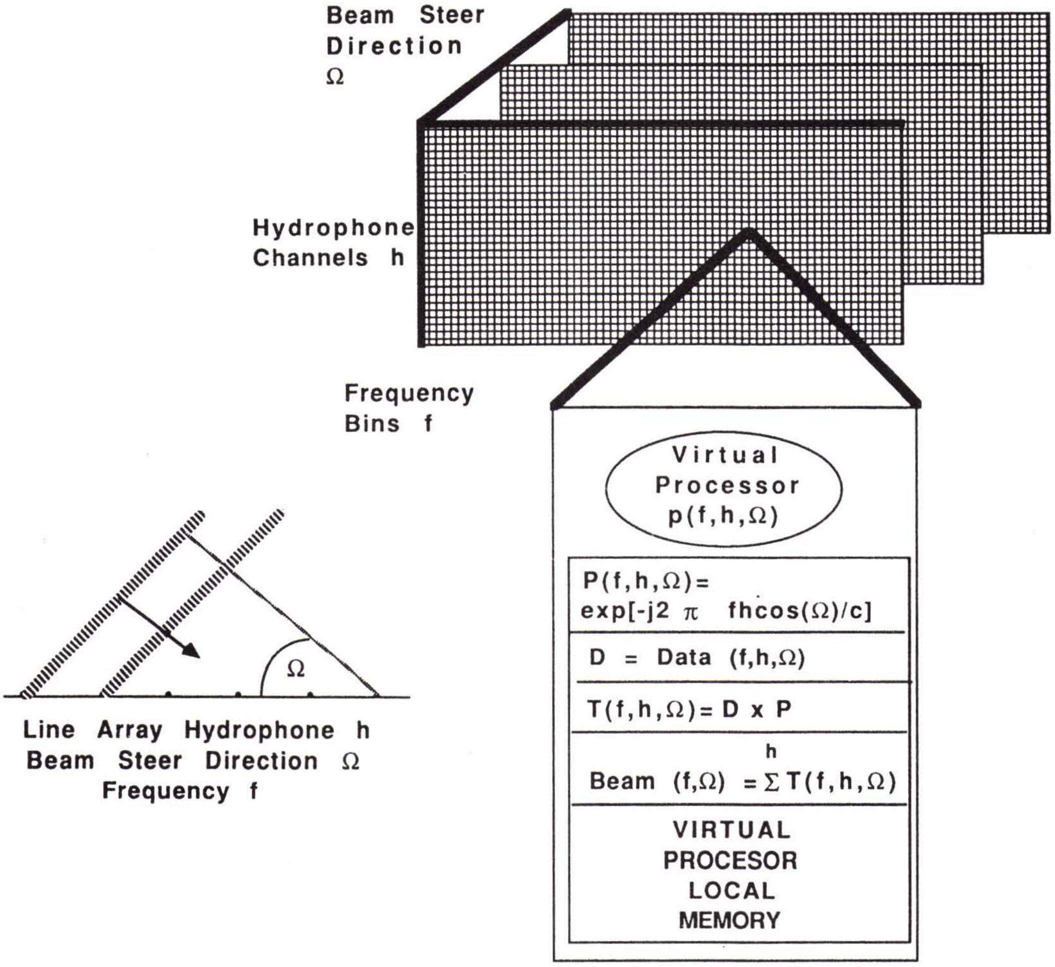

Figure 3: CM Partitioning for a SIMD Towed Array Phase Shift Beamformer

A three dimensional shape of virtual processors are configured along axes corresponding to frequency f, hydrophone position h and beam steer direction Ω p(f,h,Ω) .

Each virtual processor precalculates, in parallel, the phase shift factor P. Input hydrophone data, a function of f and h, are loaded into the appropriate virtual processors. This data is common about the beam steer Ω axis. The phase shift is applied simultaneously by all processors by SIMD multiplication of the phase shift weights to the frequency representation of the hydrophone data. Phase shifted data T is then summed about the h axis. The result are Ω beams represented in the frequency domain.

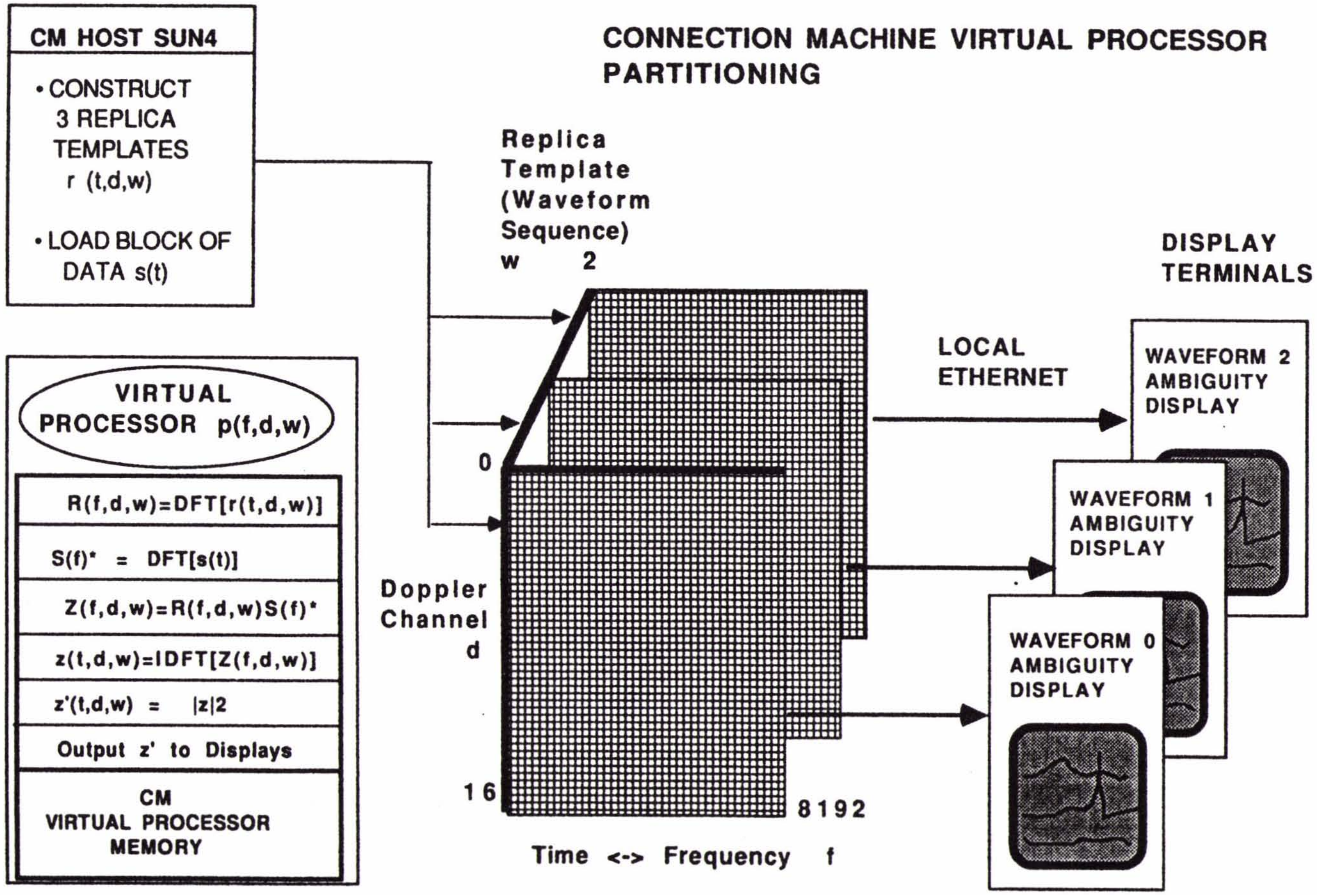
CM HOST SUN4
• CONSTRUCT 3 REPLICA TEMPLATES r (t,d,w)
• LOAD BLOCK OF DATA s(t)
VIRTUAL PROCESSOR p(f,d,w)
R(f,d,w)=DFT[r(t,d,w)]
S(f)* = DFT[s(t)]
Z(f,d,w)=R(f,d,w)S(f)*
z(t,d,w)=IDFT[Z(f,d,w)]
z'(t,d,w) = |z|2
Output z' to Displays
CM VIRTUAL PROCESSOR MEMORY
CONNECTION MACHINE VIRTUAL PROCESSOR PARTITIONING
Replica Template (Waveform Sequence)
w 2
0
Doppler Channel
d
16
8192
Time <-> Frequency f
LOCAL ETHERNET
DISPLAY TERMINALS
WAVEFORM 2 AMBIGUITY DISPLAY
WAVEFORM 1 AMBIGUITY DISPLAY
WAVEFORM 0 AMBIGUITY DISPLAY

Figure 4 CMASP Multi-Waveform Decomposition For 1 Beam

For every tranmitted waveform at every beam direction, frequency domain representations of the input data are SIMD complex multiplied to precomputed replica templates. Magnitude squared of the inversely transfomed data yield the ambiguity surfaces.

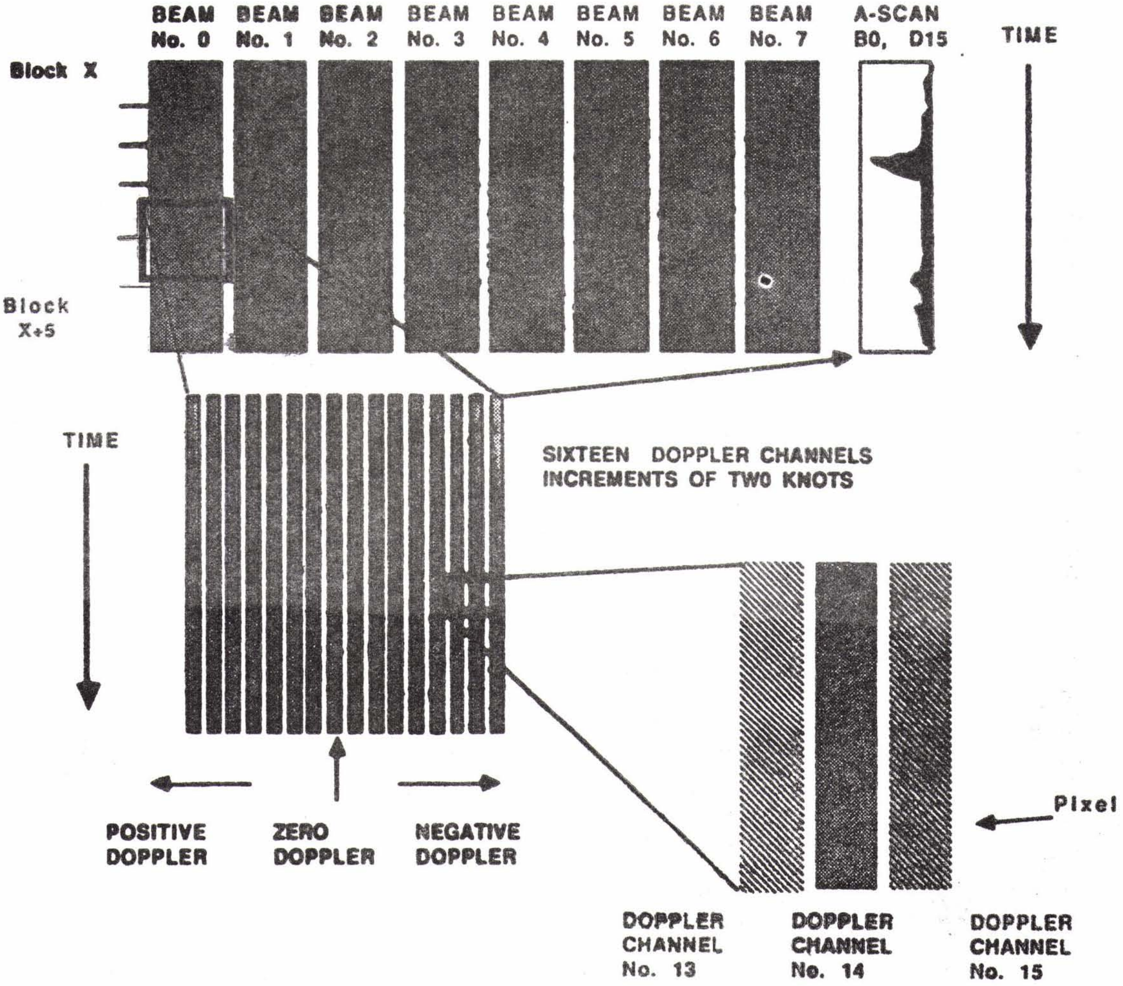

Figure 5: Connection Machine Display

The results for each transmitted waveform are displayed on a dedicated display. At each display device, for multiple beams, individual doppler channels are color-quantizied to display replica correlator outputs as a function of time

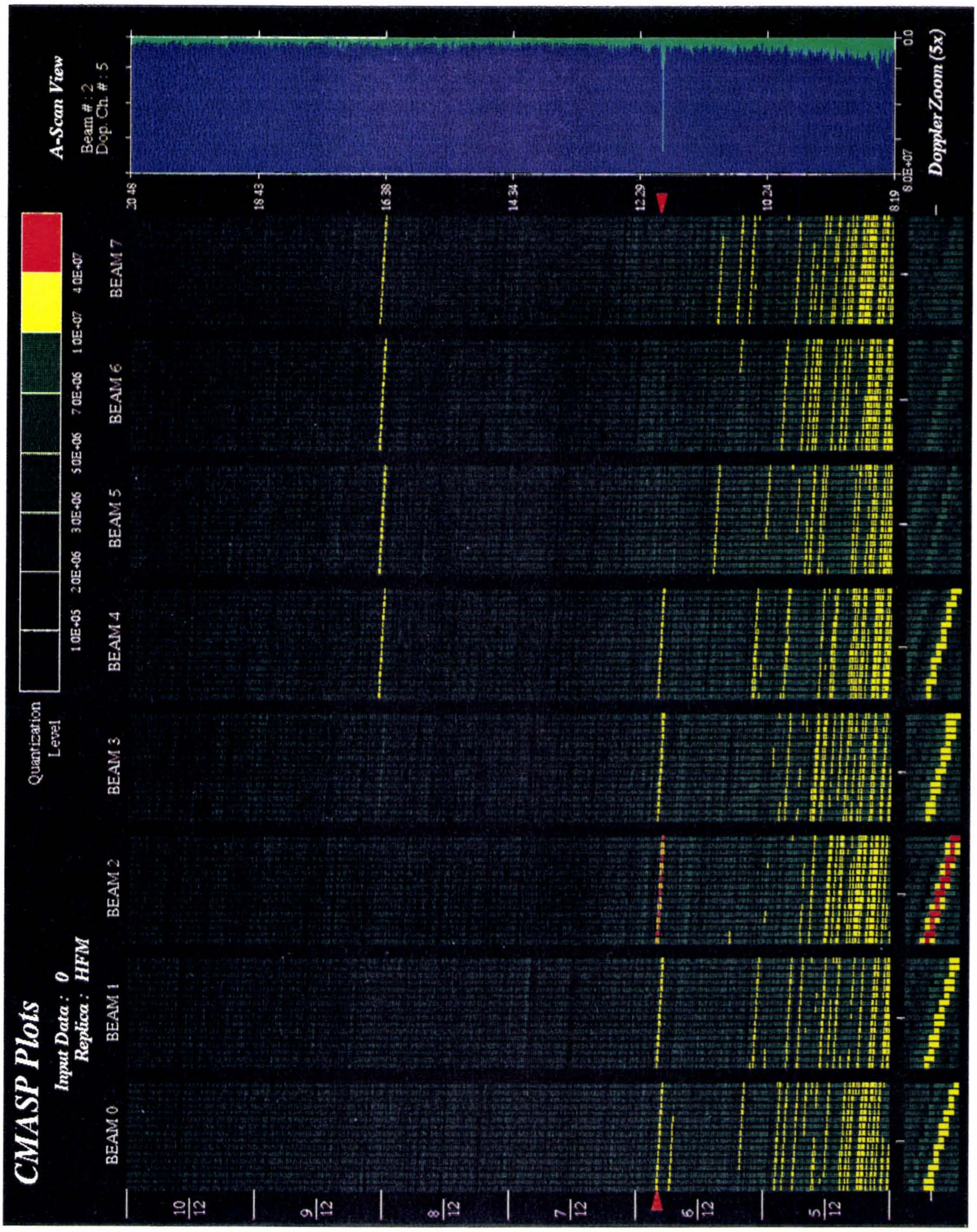

Figure 6: Connection Machine Ambiguity Surface Processor (CMASP) output for a HFM waveform. Eight beams with sixteen Doppler channels per beam are matched filterd in parallel. Waterfall time is along the vertical axis.

intentionally blank page

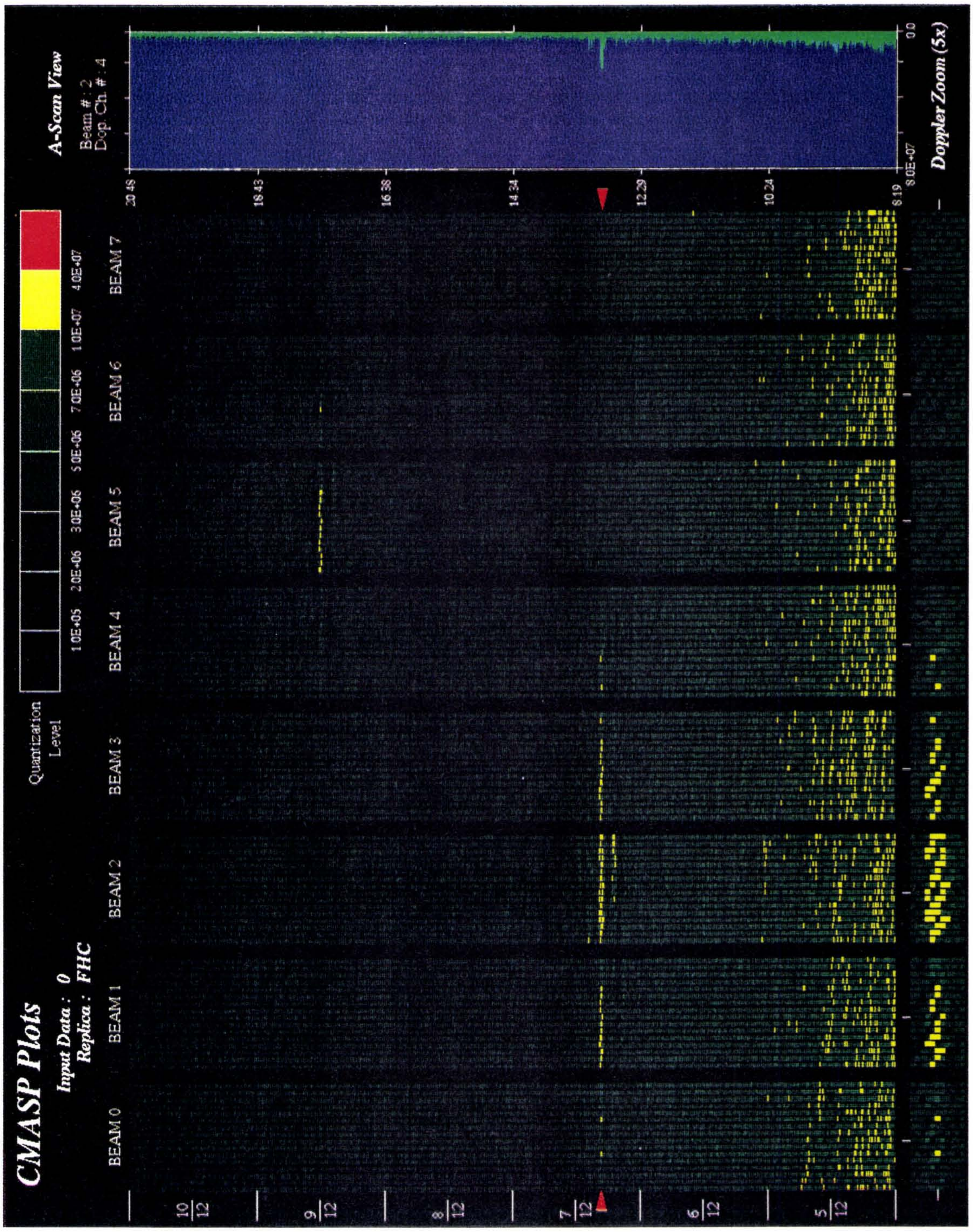

Figure 7: CMASP output for a frequency hop code (FHC).

intentionally blank page

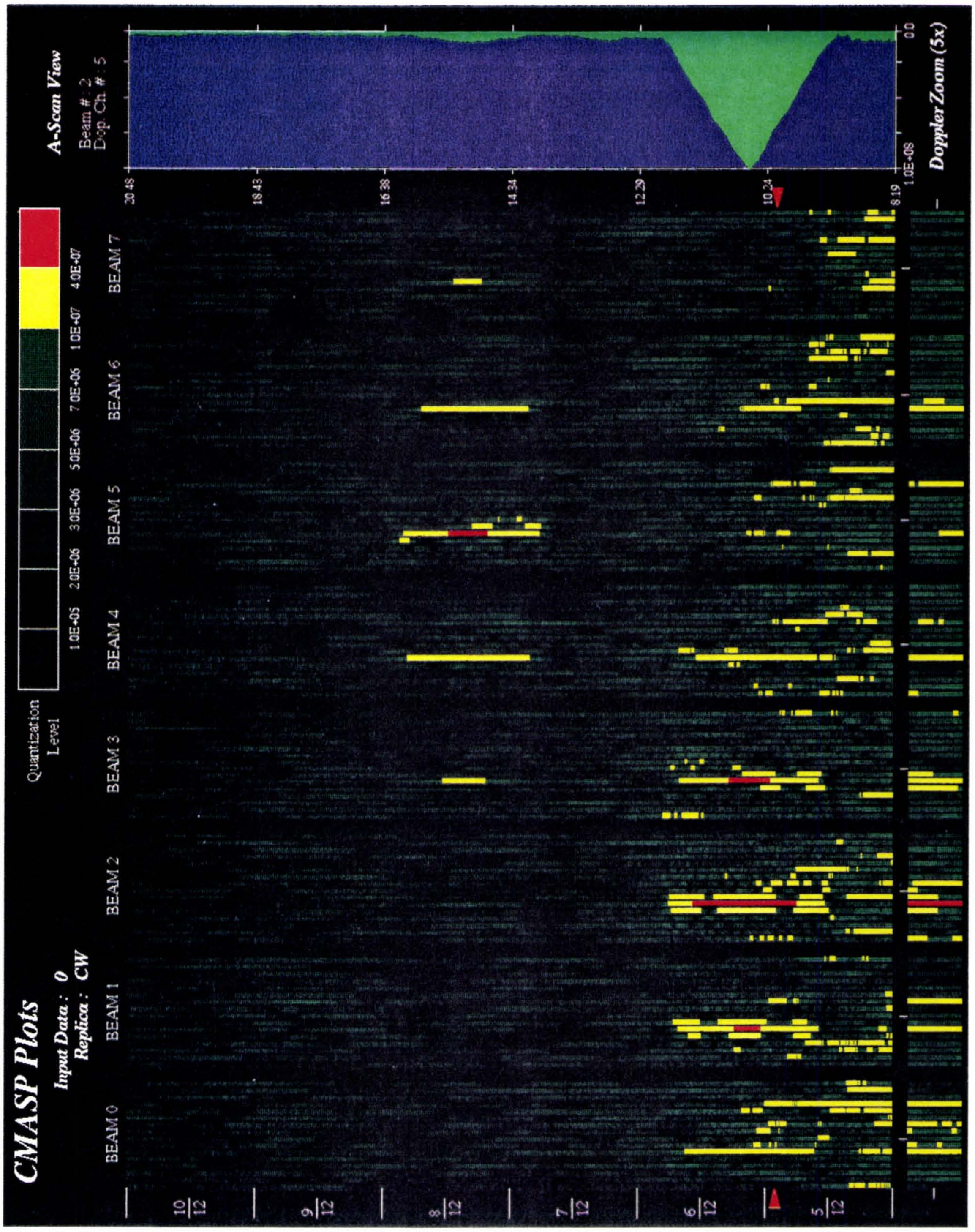

Figure 8: CMASP output for a CW pulse

intentionally blank page

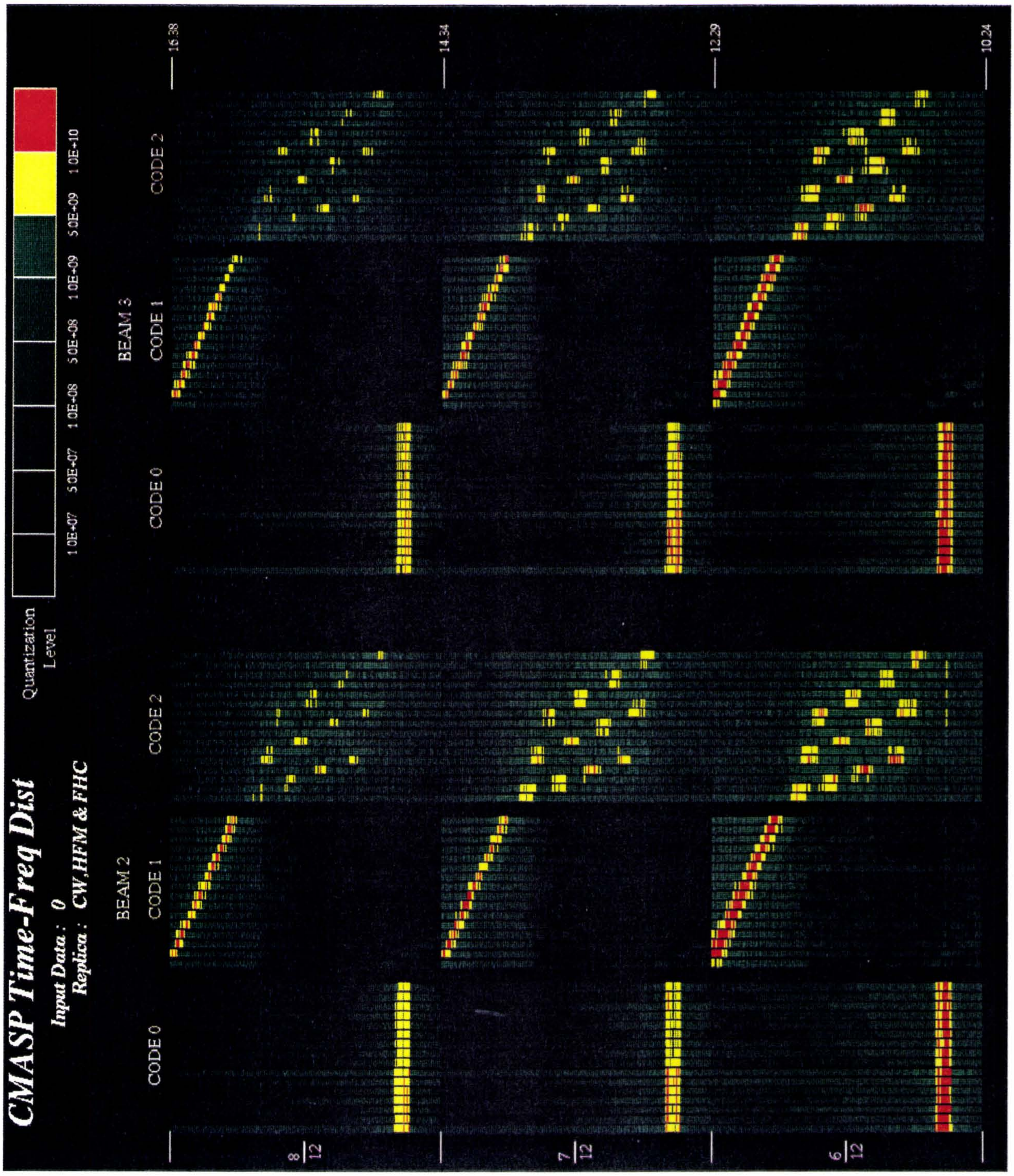

Figure 9: CMASP output for instantaneous-like time frequency processing for CW, HFM, FHC waveforms for two beams. Three block processing intevals are shown.