

ARCHITECTURAL AND PRACTICAL ISSUES IN SONAR SYSTEM PROTOTYPING

M J D Bishop

Defence Research Agency, Bingleaves, Weymouth, Dorset, UK

1. Introduction

The prototyping of Sonar systems requires fast, small, stable, available and affordable hardware elements, complemented by libraries of modular software and a comprehensive support infrastructure. This paper addresses the architectural and practical issues of providing such components for hard real time Sonar systems, operating at sampling rates of ~96 kHz.

In a Software Sonar the ordinal tasks are: analogue to digital conversion (ADC), recording, digital signal processing (DSP) and digital to analogue conversion (DAC). The interface requirements of the ADC, recording and DAC elements strongly influence the architecture of DSP systems. This paper considers the architectural requirements for (re)usable DSP building blocks, presents a range of prototype systems, and describes three DSP building blocks optimised for sonar applications which exemplify the architectural issues.

As a minimum, architectural features are required which support sampling rate conversion (typically by PLL), input-output decoupling (by FIFO or swing buffer), host control (over GPIB or VME), inter-processor synchronisation and communication, and low-distortion analogue output. Ways of meeting these requirements, and the comparative merits of alternative mechanisations are reviewed.

2. Software Sonar : Ordinal Tasks and Architectural Challenges

The ordinal tasks of a software sonar, at their simplest, are those of any sonar. A major attraction of software sonar is its ability quickly to provide real-time implementations of these ordinal tasks from readily reusable hardware and software modules. However, the architectural challenges posed by software sonars are unique consequences of its enabling technology, programmable Digital Signal Processors, and the generality demanded of the modules. We review the ordinal tasks, and identify the architectural challenges which arise.

2.1 Analog to Digital Conversion (ADC)

Multi-channel sampling, with frequency coherence (to achieve a common cadence) or phase coherence (to eliminate sample skew), is a common requirement. In a modular system, phase lock loops (PLLs) using a slow reference clock (to provide cadence) and a common reset signal (to provide synchrony) can be used to generate the high-rate clocks required by sigma-delta convertors. The centralised generation of timing information, its pragmatic distribution and the use of PLLs to generate the clocks required in each module is a common architectural mechanism in software sonar.

2.2 Non-acoustic Data

Non-acoustic data (time stamps, azimuth, transmit epochs, etc) has to be merged with the acoustic data stream prior to recording, which, as typical recording systems provide little architectural support, is rarely trivial; for an exception see [1].

ARCHITECTURAL AND PRACTICAL ISSUES IN SONAR SYSTEM PROTOTYPING

2.3 Recording

To minimise system complexity, data should flow from the sensors through the recording system to the processing system [1, 2]. This paradigm is used in the low- and high-end examples presented in the next section (figures 1-6). The alternative is an excessively complex system infrastructure.

2.4 Digital to Analog Conversion (DAC)

If phase and amplitude distortion are to be minimised the reconstruction of low-distortion analog signals requires significant oversampling, above the Nyquist rate, prior to digital to analog conversion. At audio frequencies, up to 25 kHz, commercial DACs which implement both oversampling and analogue reconstruction filters are available (for example, [3]). At higher frequencies both oversampling and analogue reconstruction require an appropriate hardware module to interpolate the data (typically by a factor of 8) and provide an accurate (typically Bessel) reconstruction filter. The key architectural challenge, essential to minimise software complexity and maximise throughput, is to decouple the input and output data streams. The A56 compact acoustic processor, described in section 4, meets this architectural requirement with an output FIFO controller which provides programable FIFO depth, automatic start-up and error detection.

2.5 Sampling Rate Conversion

Requirements for sample rate conversion arise from oversampling for digital to analogue conversion, changes in signal format (for example, from passband to single-sideband), etc. Architectural support for this capability is essential for real-time general purpose DSP modules, for example, support for multiple reference clocks and a separate N/M ratio for each output cadence. To our knowledge, this capability is not available on any proprietary DSP module.

2.6 Filtering and Detection

Sonar processing typically divides into filtering and detection operations. Filtering operations are usually performed at a small granularity (typically that of the decimation rate) and generally produce a data stream at the minimum rate for the required bandwidth. In contrast, detection operations are usually performed at a large granularity (commonly an FFT blocklength). Consequently, buffering is invariably necessary between the time series source, DSPs and the data sink. While the most apposite mechanisation changes as DSPs evolve, good support for buffering remains an essential architectural feature.

2.7 Control

Control of DSPs is best kept as simple as possible. While DSPs are not only fast and provided with sophisticated computational resources, they are however invariably equipped with only rudimentary program control capabilities. Consequently, static resource allocation is desirable with control best effected by a single master over VME, GPIB, or a similar bus. Deriving the system's cadence and buffer control from the data flow is generally a decentralised way of implementing a robust control strategy. The canny allocation of DSP resources is a recurrent challenge.

2.8 Synchronisation

Synchronisation with external events (such as a sonar's transmission epoch) and between processors is a common requirement. While DSP support for event recognition (using on-chip counters) and message passing (using mailboxes) has markedly improved, metastability remains a bogey which requires robust hardware design and defensive programming.

ARCHITECTURAL AND PRACTICAL ISSUES IN SONAR SYSTEM PROTOTYPING

2.9 Input and Output (IO)

Making IO operations affordably swift is a challenge to the system architect. Considerable time can be absorbed in bus arbitration or even in writing to a FIFO. The achievement of good trade-offs between IO cycle times and hardware complexity remains of critical importance when designing hardware modules. For example, whereas a high rate DAC must be accessible with zero wait states and possess a back to back cycle capability, control registers can typically be accessed with many wait states, as long as their access is not effected in a clumsy manner.

3. Exemplars of Prototype Systems

Rapid implementation of sonar system prototypes is self-evidently assisted by the use of off the shelf hardware. However, to effectively address the architectural challenges noted in the preceeding section, both hardware which provides the architectural hooks and modules of software which mechanise the standard DSP functions are required. The prototype systems which follow are based on a joint hardware and software building block approach and embody the components described in subsequent sections.

3.1 "Low End" Systems

Typically low end systems have a single data stream, generally recorded at 16 b x 96 kHz by a DAT recorder and are supervised by a PC. For example: an echo repeater (figure 1, [2]), instrumentation for sonar calibration (figure 2, [2]), an underwater telephone transmitter and receiver (effectively an echo repeater with a human in the loop), a pulse generator (typically already present in the echo repeater), a heterodyne and lofar analysis system (figure 3) and a multi-channel temporal signal generator for system testing (figure 4).

The building blocks used in such systems are A56 compact acoustic processors, DAT recorders, PCs and common software modules. The supporting infrastructure is generally a multi-purpose Quad A56 Box (QAB), suited to rapid reconfiguration and constant reuse.

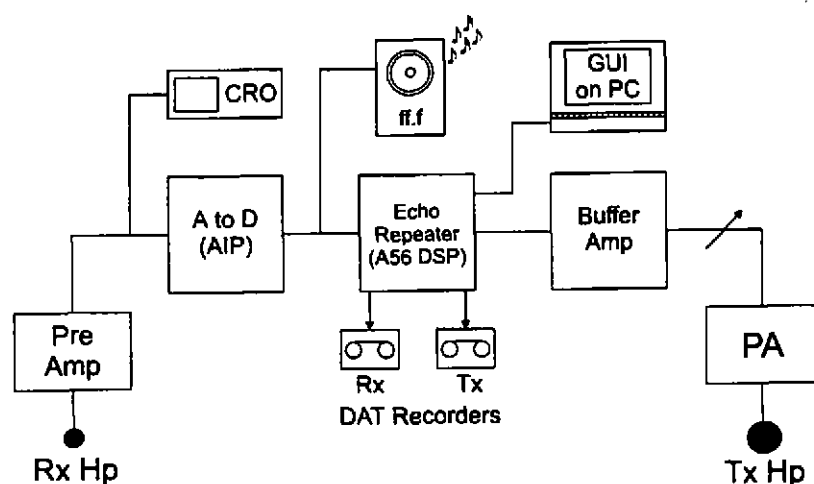


Figure 1 Echo Repeater System

ARCHITECTURAL AND PRACTICAL ISSUES IN SONAR SYSTEM PROTOTYPING

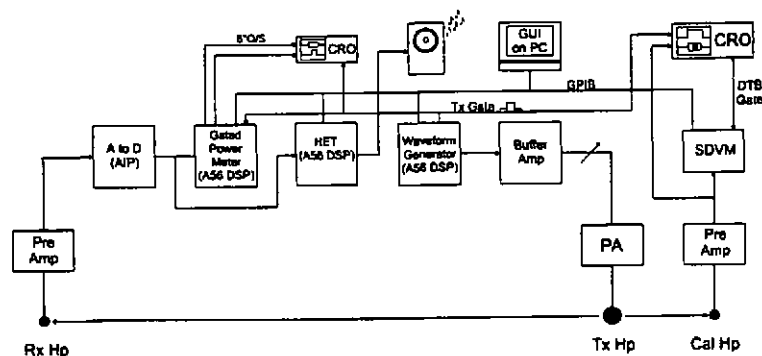


Figure 2 Echo Repeater Calibration

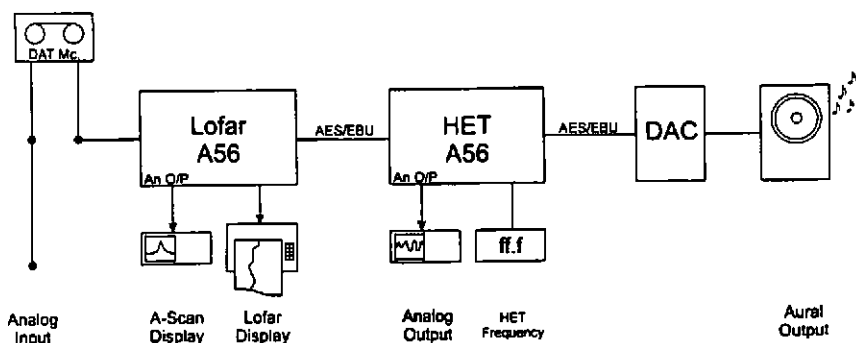


Figure 3 Heterodyne and Lofar Analysis System

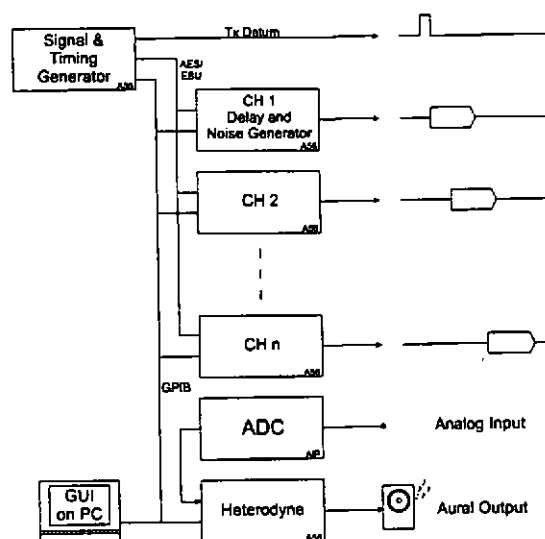


Figure 4 Temporal Signal Generator

ARCHITECTURAL AND PRACTICAL ISSUES IN SONAR SYSTEM PROTOTYPING

3.2 “High End” Systems

In contrast high end systems record and process many data streams, typically recorded via a SIF [1] on an Ampex DCRSi and supervised by a high end workstation. For example: beam and element level torpedo sonar recording and processing systems (figures 5 and 6, respectively).

The building blocks used in these systems are the D96 and S96 versatile sonar processors, described in section 5, SIF / DCRSi recording systems, workstations and multi-channel versions of the common software modules. The supporting infrastructure is generally a SIF chassis, again suited to easy reconfiguration and reuse.

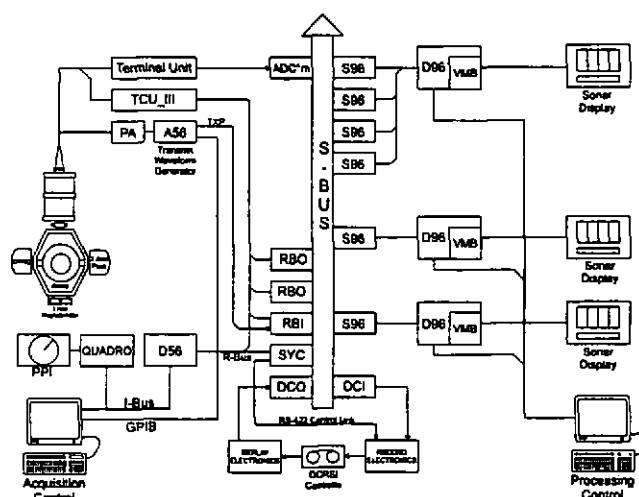


Figure 5 Beam Level Torpedo Signal Processing System

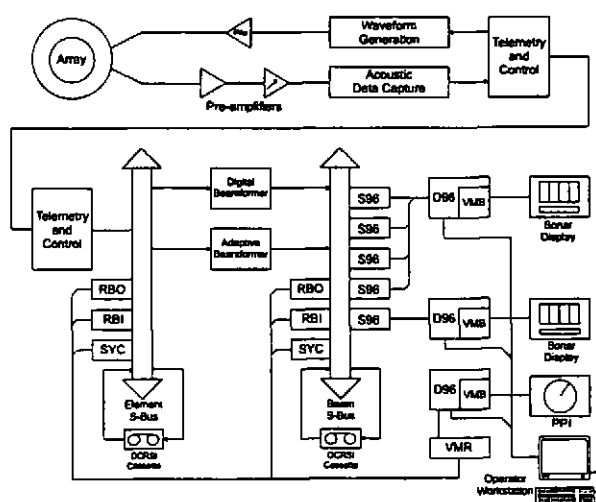


Figure 6 Element Level Torpedo Signal Processing System

ARCHITECTURAL AND PRACTICAL ISSUES IN SONAR SYSTEM PROTOTYPING

4. The A56 : Compact Acoustic Processor

The A56 Compact Acoustic Processor (figure 7, [6-9]) is a single processor DSP module with abundant input-output (IO) resources optimised for applications which span the digital - analogue interface. The A56's resources permit the bidirectional exchange of digital audio data, low distortion analogue output, discrete event IO, RS232 communication, GPIB support and provide hooks for customised interfaces. The A56 is a 3U VME P2 compatible module which can be used standalone or in a VME chassis.

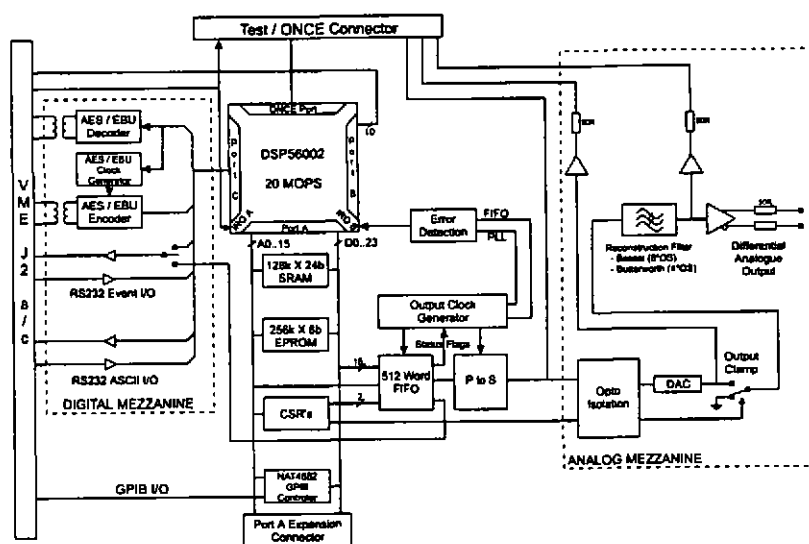


Figure 7 A56 : Compact Acoustic Processor

4.1 The Processor Core

The Motorola 24-bit fixed point DSP 56002 [4, 5] is the A56's engine. Current A56s use 40 Mhz parts delivering 20 MOPS performance. Announced members of the 56300 family will deliver up to 100 MOPS performance, and the A56 architecture is well suited to such an upgrade. The processor core encompasses the DSP, its SRAM (128 k words) and the buffers between core and peripheral (not shown in figure 7) which isolate the core from its wait stated peripherals.

4.2 AES/EBU (Digital Acoustic Data) Interfaces

Separate receive and transmit AES/EBU interfaces [3] provide DAT recorder compatible digital IO at 16b (96 dB) resolution (96dB dynamic range) and sampling rates up to 96 kHz. Using the A56 paddle board (APB, [10]), complex arrangements for AES/EBU distribution can readily be implemented, enabling rapid realisation of applications such as those described in section 3. The AES/EBU interface also supports the transport of discrete bits, one of which (the user bit) is sent with every sample and can provide a channel for an event signal.

4.3 Analogue Output

Low distortion analogue output requires a high sampling rate, careful isolation of the analogue elements and a minimal reconstruction filter. The high oversampling factors required by the analogue reconstruction filters, 8 times for an 8-pole Bessel filter and 4 times for a 6-pole Butterworth, require a low-overhead interface between the DSP and DAC if system performance is not to be adversely affected by output rates of up to 768 kHz. The FIFO memory which buffers the DAC from the DSP supports system performance both by decoupling the DSP from the output of

ARCHITECTURAL AND PRACTICAL ISSUES IN SONAR SYSTEM PROTOTYPING

individual samples and by decoupling the DSPs input and output data streams. A zero wait state, back to back cycle capable interface between DSP and FIFO is also an essential contributor to minimum overhead output. To ensure that output latency can be tuned to the task at hand a programmable and fully instrumented FIFO controller regulates the output of data.

4.4 Cadence Generation / Modification capabilities

Signal processing algorithms frequently have distinct input and output rates, typically related through processes such as decimation or interpolation. While the input and output clocks used in a processing system can be centrally generated, skew between clock and data paths can make this approach problematic. To avoid these problems and the attendant complexity in the applications infrastructure, the A56 uses phase locked loops to generate the clocks required for the analogue output and AES/EBU transmitter from a cadence derived from either the AES/EBU receiver, an on-board crystal or a backplane input. This approach provides full flexibility, while minimising complexity.

4.5 System Integration Interfaces

While the A56 can be applied to dedicated (EPROM bootstrapped) applications, its design provides hooks for its integration into wider systems. The GPIB talker, listener, controller capability provided by the NAT 4882 controller enables A56s to be interfaced with a wide range of computers, facilitating their use as elements of a wider system. The echo repeater calibration system, figure 2, is a good example of several A56s used as (down line loaded) instruments. Bidirectional discrete event IO, which can be sourced from the output FIFO to eliminate any timing skew between acoustic and event outputs and a duplex RS232 interface complete the high-level interfaces.

Low-level bit IO interfaces are provided on the P2 connector which can implement slot address sensing, for auto configuration, or simple digital IO, to pass parameters or data. Potential for system expansion is provided by the port A expansion connector which permits secondary daughterboards to be interfaced to the DSP core.

4.6 Support and Test Interfaces

Software debugging is performed using the standard Motorola CLASA and ADSA tools, and the 56002s OnCE port which is accessed via the A56s test connector, fitted to the A56s front edge. Also accessed via the test connector are numerous key signals and data paths which are available for design validation and system integration. The A56 test connector provides access to FIFO and other status bits, event IO, RS 232 traffic and the digital acoustic data streams. The Operational Test Box (OTB_2, figure 8) reformats this data for assimilation by the human operator and buffers it for input to test instruments.

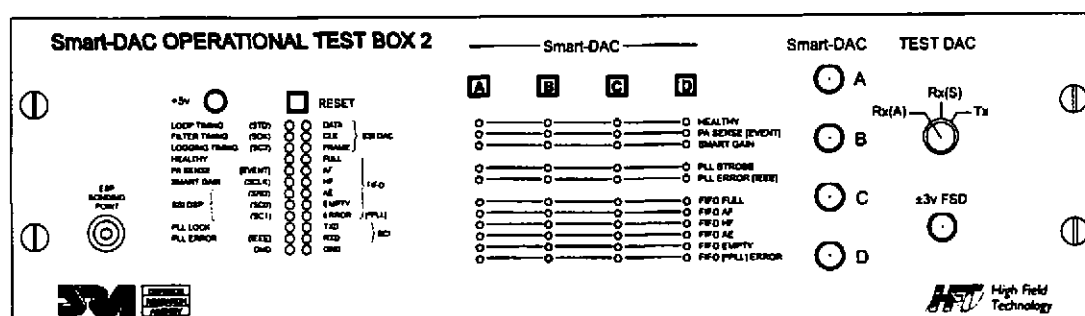


Figure 8 Operational Test Box : Front Panel

ARCHITECTURAL AND PRACTICAL ISSUES IN SONAR SYSTEM PROTOTYPING

5. The S96 and D96 : Versatile Sonar Processors

The S96 [11] and D96 [12] versatile sonar processors (figures 9 and 10 respectively) are dual-processor DSP modules optimised for multi-beam sonar applications. The S96 provides a powerful engine for filtering and detection tasks by providing an S-bus hosted M-bus slave which can fetch acoustic, non-acoustic and event data from the S-bus, deposit results in the slow SRAM for collection by a D96, input and output aural or acoustic signals from and to AES/EBU, and output onto the S-bus. The D96 complements the S96 by providing a VME hosted M-bus master which can fetch data from one or more S96s for formatting, display on the D96's video mezzanine board (VMB [13], figure 11) and pass it to a host over the VME bus. Alternatively, using the same dual-processor core as the S96, the D96 can act as a single channel sonar processor, using its AES/EBU receivers and transmitters to input and output data.

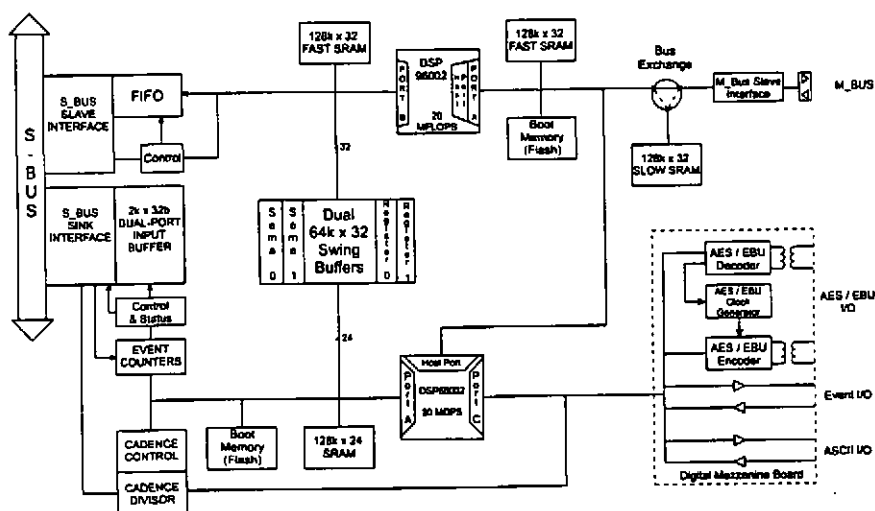


Figure 9 S96 : Versatile Sonar Processor

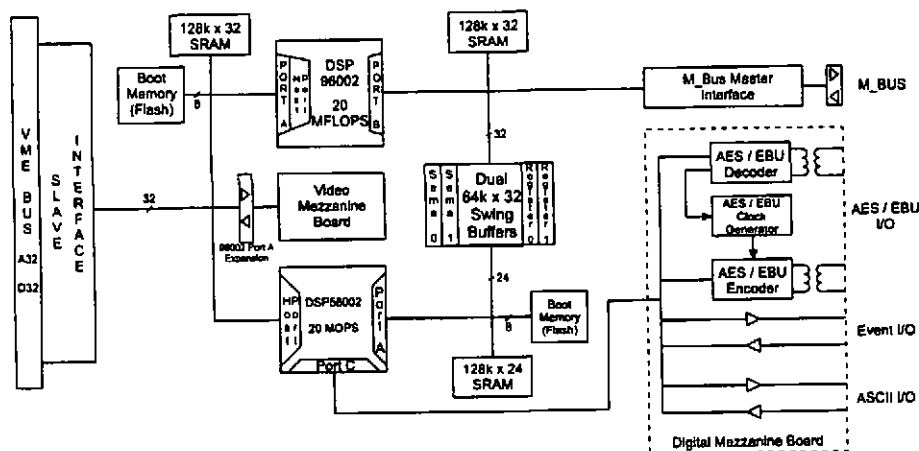


Figure 10 D96 : VME Sonar Processor

ARCHITECTURAL AND PRACTICAL ISSUES IN SONAR SYSTEM PROTOTYPING

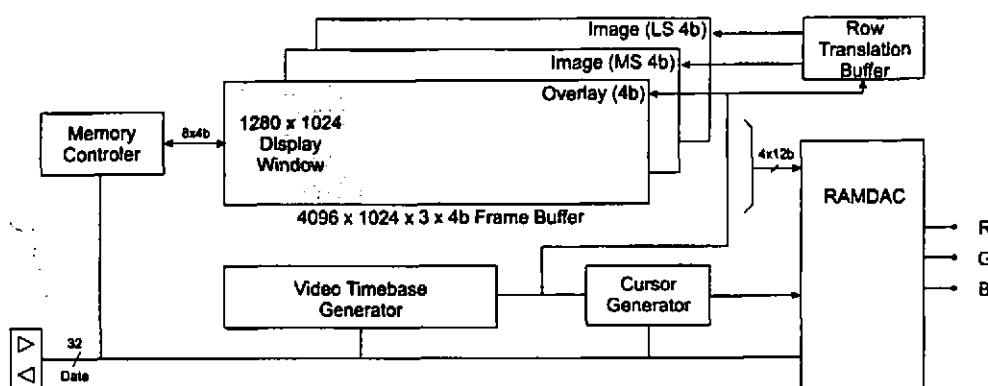


Figure 11 VMB : Video Mezzanine Board for D96

5.1 The S96/D96 Dual-Processor Core

The Motorola 24-bit fixed point DSP 56002 provides the filtering engine. Like the A56, the dual-processor core uses 40 MHz parts which deliver 20 MOPS performance and 128k x 24b of SRAM divided between the X, Y and P memory spaces. The Motorola 32-bit floating point DSP 96002 [14] provides the detection engine, again using 40 MHz / 20 MOPS parts, with on each of its external memory busses 128k x 32b of SRAM for the rapid execution of large FFTs. A 64k x 32b dual swing buffer provides the interface between the DSPs. To effect the 24 to 32 bit transition the swing buffer has sign-extension, zero-packing and double-word access capabilities on its 56002 side; to enable the implementation of synchronisation and event primitives, it has semaphore and communications registers. Both processors boot memories are implemented in Flash, which can contain either standard self-test / secondary bootstrap code or application specific code for deeply embedded applications, and can be reprogrammed in-situ.

5.2 The S96 S-Bus Interface

The S-Bus provides the S96 with its data through a dual port input buffer, which supports four simultaneous streams of acoustic, non-acoustic and recorder status data. Also, the event counters which mark the transmit epochs and the cadence clock which provides a frequency reference for reconstructed analogue data are derived from the S-Bus. Processed results or data (for example, transmit waveforms) for recording on the DCRSi can be sourced onto the S-Bus through a FIFO.

5.3 The S96 M-Bus Slave Interface

The S96's M-Bus slave interface provides the M-Bus master, typically a D96, with non-blocking access both to the slow SRAM, and to the host ports of both processors, permitting remote initialisation and control of the S96. Results from the S96 are typically staged in the large slow SRAM, which (using a three-port bus-exchanger) provides non-blocking access by either the DSP 96002 or the M-Bus.

5.4 The D96 M-Bus Master Interface

The M-Bus master interface provides the D96 with supervisory and data access to its S96 slaves. Typically, figures 5 and 6, one D96 supervises one or more S96s and displays their normalised data.

5.5 The D96 VME-Bus Slave Interface

The D96's VME-Bus slave interface provides the workstation or embedded VME controller with access to D96 and S96 resources and data. Typically, several D96s and their S96 slaves are controlled by a single workstation, figures 5 and 6.

ARCHITECTURAL AND PRACTICAL ISSUES IN SONAR SYSTEM PROTOTYPING

5.6 The D96 Video Mezzanine Board (VMB)

The VMB provides a closely coupled graphics resource for the D96 and its slave S96s. The large multi-nibble framestore, figure 11, supports the straightforward display of a significant volume of Sonar data. Hardware support for scrolling and cursor generation minimise wasted host cycles and demands for framestore bandwidth. Architecturally, the framestore can be accessed as memory by both the DSP 96002 and the host processor enabling the distribution of tasks between the processors.

6. Sharc Building Blocks and Their Application

The Analog Devices SHARC [15] is a recently available floating point DSP providing 40 MOPS performance. The generic SHARC VME board (figure 12) is available, with some variation, from a number of vendors. One of the SHARCs strengths is its multiprocessor capability, which permits six processors to share a multi-master bus and a common memory with a host processor. Additionally, the SHARC has serial ports and inter-processor links for static communication paths. The mezzanine interface site could host ADCs and DACs, or provide a master or slave bus interface.

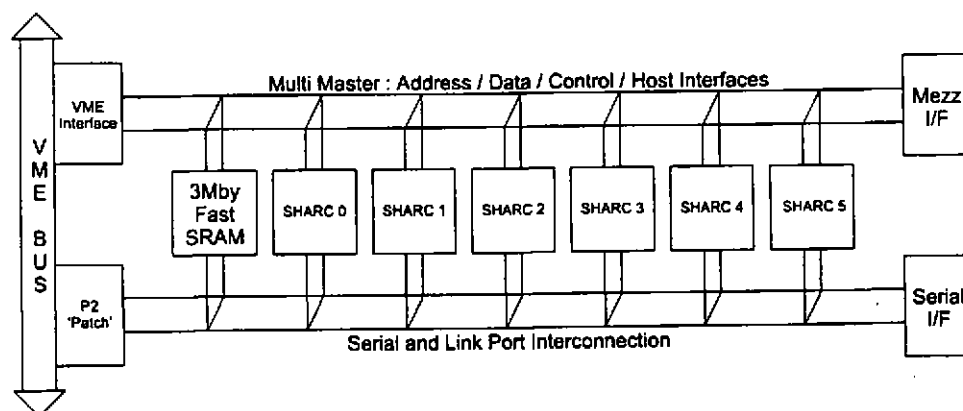


Figure 12 Generic SHARC VME Board

The SHARC could replace the processor core of either the S96 or D96. However, as Motorola have announced a 100 MOPS DSP 56301, the substitution of the DSP 96002 by several SHARCs may be preferable. The similarities between the architecture of the processor core, with an overt dual-port RAM and semaphores, and a pack of SHARCs with internal dual-port RAM suggest that it is only implementation not architecture which has changed. Also, the bulk of the S96 design is IO related and effectively processor independent.

The implementation of signal processing systems using COTS VME / SHARC boards is attractive. With an M-Bus slave interface implemented on the mezzanine site, a generic VME board can straightforwardly receive data from the DCRSI/SIF via an S96 and output its data to the D96 controller or to a VME based host. Again building on the Sonar oriented architectural features of the S96 and D96.

ARCHITECTURAL AND PRACTICAL ISSUES IN SONAR SYSTEM PROTOTYPING

7. Software Building Blocks

Software applies hardware to a specific task and makes it useful. However, DSP software is especially difficult to craft. Consequently reusable software building blocks are extremely attractive. The A56 has a GPIB server, resident in PROM, which provides loader, peek and poke, and RS 232 / GPIB IO drivers. Signal processing software building blocks for A56, S96 and D96 processors provide implementations of a wide range of signal processing functions, including:

- interrupt driven IO for AES / EBU Rx and Tx operations;
- decimation chains for 96kHz to 6/4/3/2 kHz bandwidth;
- interpolation chains for 2/3/4/6 kHz bandwidth to 96 kHz;
- four and eight times oversampling interpolators for 96 kHz input;
- noise generators (decimated one bit PRBS);
- pulse generation logic for swept, shaded and continuous phase FSK pulses;
- power calculation and automatic gain control widgets;
- compensation filter widgets for narrowband and wideband applications;
- FFTs and normalisation for spectrum analysis;
- correlators and normalisation for pulse compression.

8. Conclusions

The rapid development of Sonar prototypes depends on the use of familiar hardware. Additionally, the available hardware must be genuinely flexible: how often are the full range of applications known when a component or system is specified? A small family of Sonar processors has been described which, hopefully, addresses these injunctions.

The (digital) signal processing aspects of Sonar system prototyping are "almost" easy, when compared with the complex infrastructures which manage and record the data in the examples of section 3. The designs of the three sonar oriented DSPs presented have emphasised interprocessor communication and input-output. While the SHARC has addressed the first requirement, with heterogeneous processor arrays, COTS DSP remains to convincingly address the IO requirement.

ARCHITECTURAL AND PRACTICAL ISSUES IN SONAR SYSTEM PROTOTYPING

9. References

- [1] M J D Bishop, 'The Architecture of Digital Homing Systems', Proc UDT Conference 1993
- [2] M J D Bishop, 'Transducer Equalisation : Signal Representation and Filter Structure', Proc IOA 1995
- [3] Crystal Semiconductor Corporation, 'Digital Audio Products Databook', January 1994
- [4] Motorola, 'DSP 56000 Digital Signal Processor User's Manual', DSP56000UM/AD Rev 2, 1990
- [5] Motorola, 'DSP 56002 Digital Signal Processor User's Manual', DSP56002UM/AD Rev 1, 1993
- [6] High Field Technology, 'Specification and User Manual for the A56 Processor', 1995
- [7] High Field Technology, 'Specification and User Manual for the A56 Digital Mezzanine', 1995
- [8] High Field Technology, 'Specification and User Manual for the A56 Analogue Mezzanine (Bessel Filter)', 1995
- [9] High Field Technology, 'Specification and User Manual for the A56 Analogue Mezzanine (Butterworth Filter)', 1995
- [10] High Field Technology, 'Specification for the APB_2', 1994
- [11] High Field Technology, 'Specification for the S96 Processor', Issue 7, 1994
- [12] High Field Technology, 'Specification for the D96 Processor', Issue 7, 1995
- [13] High Field Technology, 'VMB Specification', Issue 2, 1995
- [14] Motorola, 'DSP 96002 Digital Signal Processor User's Manual', DSP96002UM/AD, 1989
- [15] Analogue Devices, 'ADSP-2106x SHARC User's Manual', 1st Ed, 1995

10. Acknowledgements

Understanding of the issues described was only achieved through interaction with many colleagues. Particularly Dr C J Bryant, Dr M J Cox and I Middleton of High Field Technology.

Any views expressed are those of the author and do not necessarily represent those of the Defence Research Agency / HM Government.

© British Crown Copyright 1995 / DRA

Published with the permission of the Controller of Her Britannic Majesty's Stationery Office

FALSE ALARM STATISTICS OF AREA-DEPENDENT REVERBERATION

Dieter Brecht, Franko Greiner, Jan-Peter Babst, Jochen Ziegenbein

Federal Armed Forces Underwater Acoustics and Marine Geophysics Research Institute (FWG),
Klausdorfer Weg 2-24, 24148 Kiel, Germany

1. DESCRIPTION OF THE PROBLEM

Low Frequency Active Towed Array SONAR-Systems (LFTAS) are used to detect submarines at long distances. A short sonar pulse is sent out and the backscattered echoes are collected. After signal processing, the received signals are analysed and existing targets are (hopefully) detected. Besides true targets, the signal amplitudes consist of surface/volume reverberation and noise. Especially in shallow water, bottom reverberation produces high signal amplitudes and therefore a high false alarm rate. A first separation of true targets from obvious false alarms can be done by defining a threshold: all amplitudes smaller than, for example, 10 db relative to the mean are rejected. The remaining fraction can be displayed as an image on a monitor, so the operator can use his visual capabilities and experience to extract target-induced patterns. The operator is able to deal with a certain rate of alarms, for example coloured pixels on a screen. If the alarm rate is too high, it can and will be adjusted by increasing the threshold. Consequently, for a given target strength the additional amount of high amplitudes generated by reverberation/noise directly decreases the detection probability of the sonar system. In order to estimate the performance of a SONAR system, the distribution of the false alarm amplitudes therefore has to be known as a function of location and system parameters.

2. EXPERIMENTAL SETUP

In the years 1992-1994 several sea trials have been carried out with our research vessel PLANET to investigate the detection performance of an experimental active towed array system (ATAS) under shallow water conditions. The receiving antenna consisted of a 64-element towed array allowing an angular resolution of 2° at broadside. The operating frequency was about 1 kHz. A special transducer allowed to illuminate the left and right side independently to avoid left/right ambiguities. The received echo signals were digitized, beamformed, matched-filtered and finally normalized independently for 60 beams ($\pm 60^\circ$ from broadside). Normalisation was done by dividing the time series by the output of a running median filter. This operation is similar to a high pass filter cancelling all structures larger than a certain distance, for example 500m. Besides the decay of the reverberation also large scale changes in bottom backscattering strength are thus eliminated from the time data.

The resulting time series of 60 independent beams are, after data reduction, put into an echogram display (time/range versus beam number), each beam containing the history of several pings [3].

3. AMPLITUDE DISTRIBUTION FUNCTIONS

For large distances, the signal amplitudes are dominated by noise. As a superposition of many uncorrelated statistical processes, for example in the water or the amplifiers, the resulting signal amplitudes are normal distributed and therefore the amplitudes of the envelope are expected to be Rayleigh-distributed:

$$f_R(x) = \frac{x}{\sigma^2} \cdot e^{-\frac{x^2}{2\sigma^2}} \quad (1)$$

FALSE ALARM STATISTICS OF AREA-DEPENDENT REVERBERATION

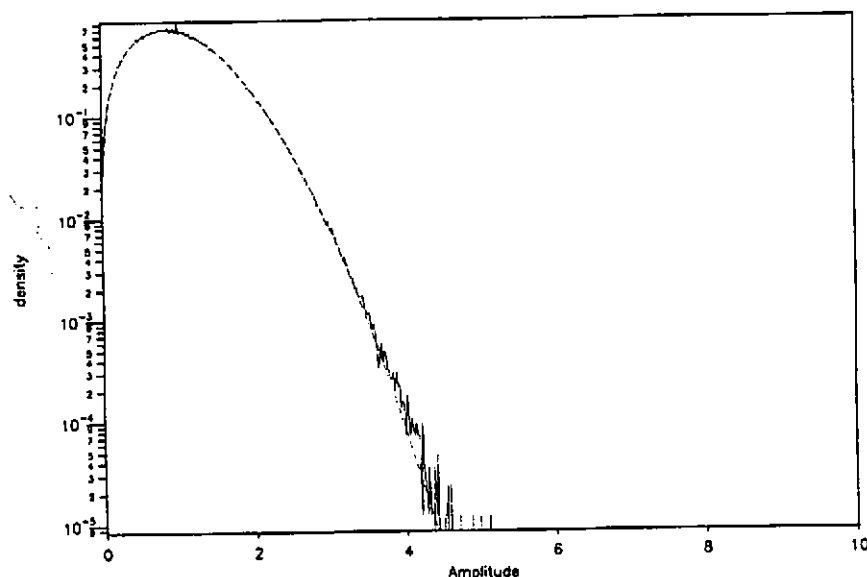


Figure 3.1: Amplitude density-function: real data ('noise limited' conditions) , fitted Rayleigh distribution (dashed line)

with $f(x)$ amplitude distribution function, x signal amplitude. Fig. 3.1 shows a distribution function of data collected during our sea trials. The data were taken from distances beyond 30km to ensure noise limited conditions. The data fit very good to the Rayleigh distribution included in the diagram. Since all amplitudes are normalized to the median (the median of the distribution is approx. 1), the shape of the Rayleigh distribution is independent from any parameter.

The influence of additional reverberation can be seen in fig. 3.2 Especially at shorter distances the reverberation component becomes more and more important. The distribution shown in fig. 3.2 is based on data collected at a range around 16 km. The resulting distribution differs significantly from Rayleigh. Especially the higher amplitudes are underestimated by the Rayleigh distribution, leading to an increased false alarm rate for a fixed threshold.

The amplitudes of purely ocean boundary-induced reverberation are often found to be distributed lognormal [1],[2]. Different from the Rayleigh distribution, the lognormal distribution

$$f_{log}(x) = \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{(\ln x - \mu)^2}{2\sigma^2}} \quad (2)$$

depends on two parameters, the mean μ_{log} and the standard deviation σ_{log} of the amplitudes' logarithm. While normalisation forces the parameter μ_{log} to be 0 ($\mu_{log} = \log(\text{median}(f_{log}))$), the second parameter σ_{log} can still vary as a function of reverberation conditions.

The comparison of a data set with the logn.-distribution can be seen in fig. 4.3. To ensure a high clutter-to-noise ratio of the amplitudes, i. e. a reverberation limited situation, the data were collected in the close vicinity of the antenna (range approx. 5 – 10 km) and (for the left graph) only the broadside beams were used to ensure

FALSE ALARM STATISTICS OF AREA-DEPENDENT REVERBERATION

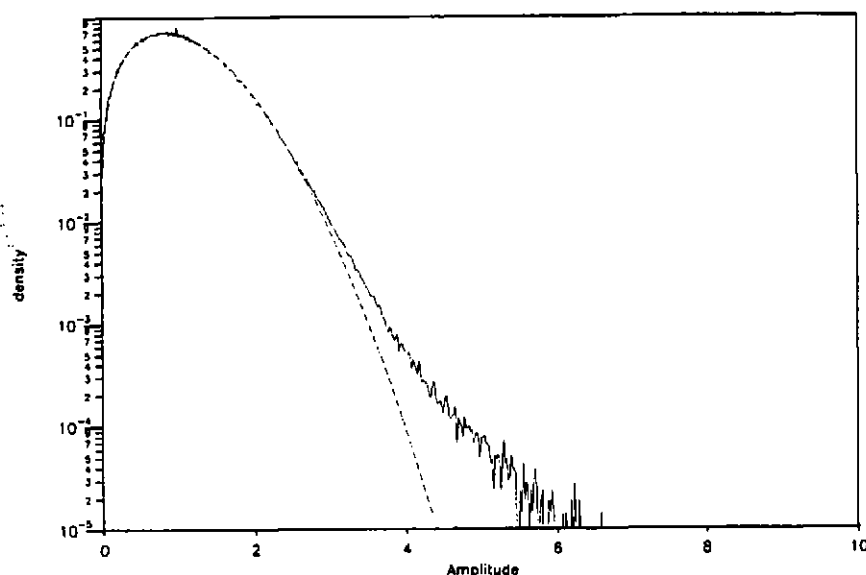


Figure 3.2: Amplitude density-function: real data, fitted Rayleigh distribution (dashed line)

constant angular resolution and aspect angle. As far as high amplitudes are concerned, the data can be described much better by the lognormal-distribution. For small amplitudes, the logn-distribution does not fit to the data and the distribution seems to be still Rayleigh-like.

The logn-functions displayed in fig. 4.3 were calculated for the parameters $\mu_{log} = 0$ and σ_{log} fitted to the data. Because the real data, especially for small amplitudes, are not really logn-distributed, μ_{log} might differ from 0. Optimizing both parameters according to the high-amplitude part of the distribution therefore improves the agreement. (see fig. 4.4: same data as in fig. 4.3, right graph). For the data in fig. 4.3/4.4 we found $\mu_{log} = 0$ and $\sigma_{log} = -1$. The parameter $e^{\mu_{log}}$ could be seen as a measure of the ratio of clutter to clutter-plus-noise, while σ_{log} is a second parameter characterizing the bottom type and structure.

For the data collected in the close vicinity of the antenna we expect reverberation limited conditions, i. e. Clutter $C >$ Noise N . This is exactly what we see if we assume $C + N$ and N to be the mean of the amplitudes before normalization, $C + N$ at the considered area and N at sufficiently large distances in order to ensure nearly noise limited conditions (fig. 4.5). As a consequence, the Rayleigh shaped distribution function for amplitudes smaller than 2 cannot be explained only by dominating noise-processes which are independent of range and reverberation level.

4. FALSE ALARM RATE

Unfortunately, the entire distribution cannot be described by an analytical function $f(p_1, p_2 \dots; x)$, depending on a few known parameters like distance, incident angle, bottom type (mean reverberation level) or noise level.

FALSE ALARM STATISTICS OF AREA-DEPENDENT REVERBERATION

parameters and the false alarm rate could have been given as a simple function $P_{FA}(x) = \int_x^\infty f(x')dx'$ of the threshold (and vice versa). This is necessary, for example, when the alarms of an area with varying distance and bottom type are to be displayed in one image with a given and constant density of alarms.

For the lognormal distribution, this transformation could be done by taking $y = (\ln x - \mu_{\log})^2 / \sigma_{\log}^2$ while for Rayleigh distributed data a simple normalization $y = x/\bar{x}$ would be sufficient.

As a first step, we found the detection index $DI = 10 \log(x - \bar{x})^2 / \sigma^2$ (x amplitude and \bar{x}, σ its mean and standard deviation) to give quite good results [4]. The following figures show the false alarm probability P_{FA} as a function of the detection index DI .

Fig. 4.6 compares data from broadside direction with data from all beams, covering an azimuth sector of $\pm 60^\circ$. The broadside beams give a lower false alarm probability, indicating an azimuthal anisotropy in backscattering statistics.

Fig. 4.7 demonstrates the influence of the bandwidth. While the other parameters (location, direction, range) were kept fixed, the used bandwidth was changed from 128 to 32 Hz. As can be seen, the probability of false alarm seems to be independent from the bandwidth. Nevertheless, we also found examples where the larger bandwidth showed a smaller false alarm rate (fig. 4.8) This will be a field of further investigations. It should be pointed out that in cases where the probability of false alarms does not depend on bandwidth, the false alarm rate will be higher for the larger bandwidth signals. This has to be taken into account when modelling the potential gain in reverberation suppression by increasing the bandwidth.

Fig. 4.9 shows an example of results which were obtained when directional transmission to starboard and port side was used. In this example, the starboard data show an increased false alarm rate as compared to port. A possible explanation may be different incident angles due to a slight bottom inclination.

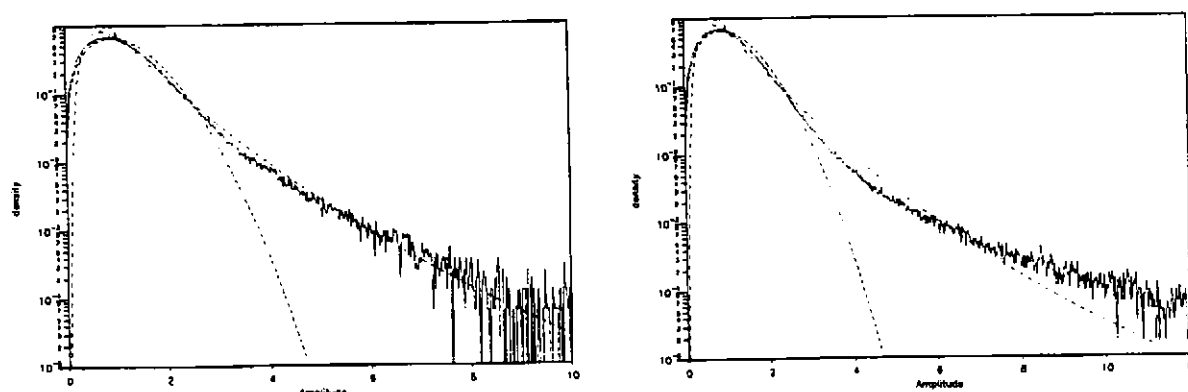


Figure 4.3: Amplitude density-function, real data compared to logn. and Rayleigh distribution. Data were collected at very short distances under reverberation limited conditions. Left graph: broadside beams only, right graph: beams covering the complete $\pm 60^\circ$ -sector. The difference indicates the azimuthal anisotropy of the backscattering statistics.

FALSE ALARM STATISTICS OF AREA-DEPENDENT REVERBERATION

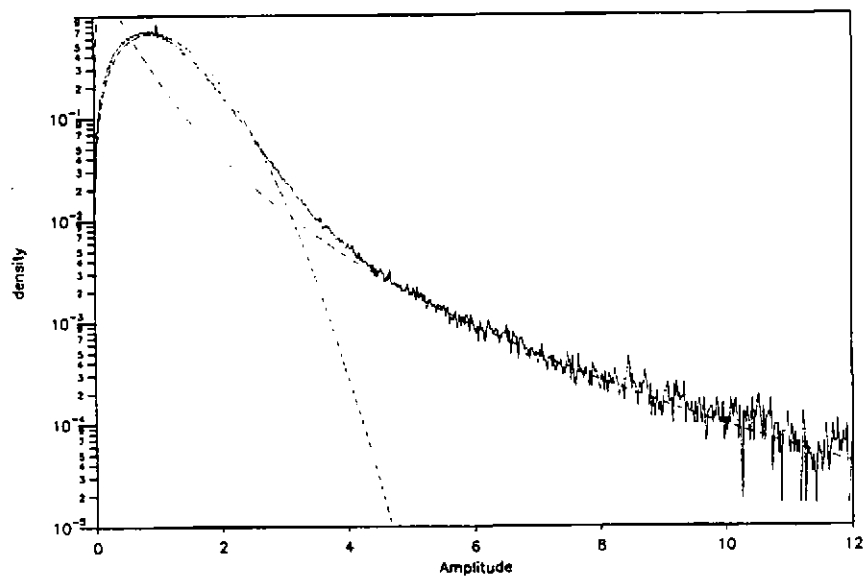


Figure 4.4: Amplitude density-function: real data (solid line), compared to fitted logn-distribution and Rayleigh-distribution. Data collected at very small distances, reverberation limited conditions.

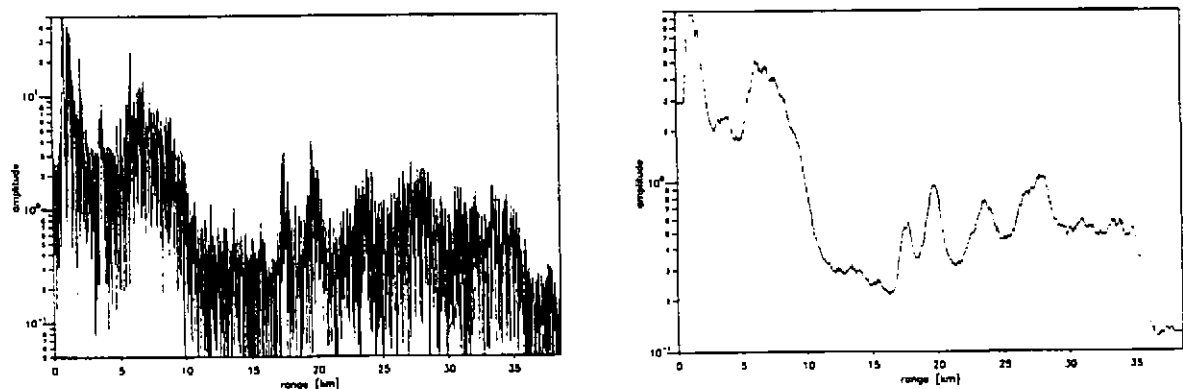


Figure 4.5: time series of a broadside beam, before and after median filtering

FALSE ALARM STATISTICS OF AREA-DEPENDENT REVERBERATION

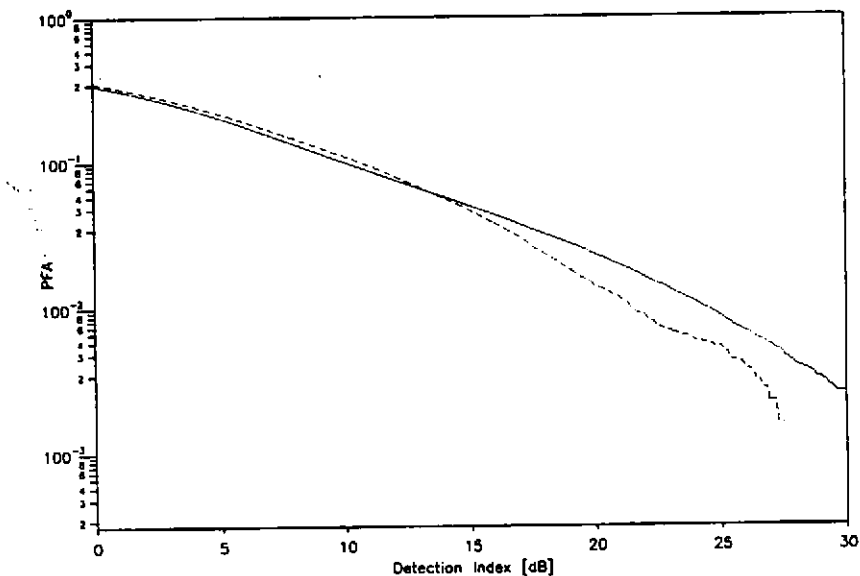


Figure 4.6: Probability of False Alarm (PFA) versus detection index (DI). Solid line: data from a 60° azimuth sector around broadside, dashed line: data from broadside only, dotted line: Rayleigh-approximation.

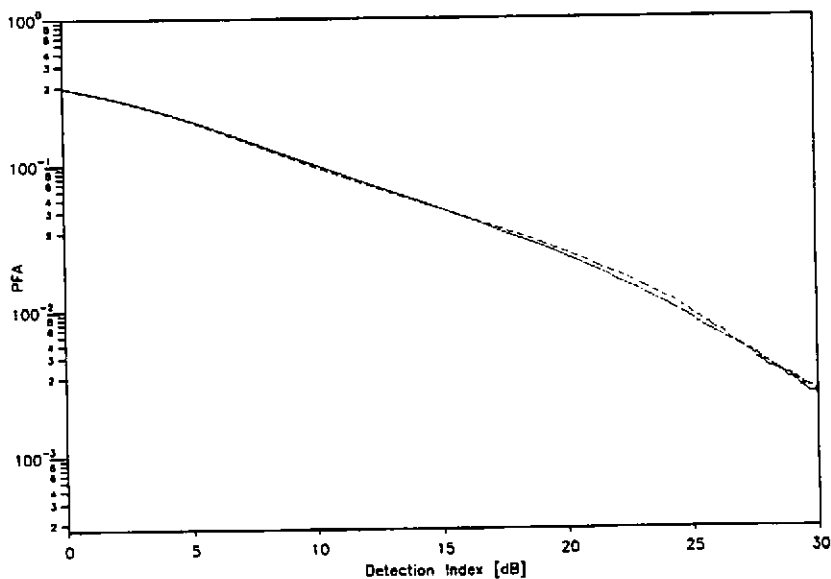


Figure 4.7: Probability of False Alarm (PFA) versus detection index (DI) for different bandwidths BW. Solid line: BW = 128 Hz, dashed line: BW = 32 Hz, dotted line: Rayleigh-approximation

FALSE ALARM STATISTICS OF AREA-DEPENDENT REVERBERATION

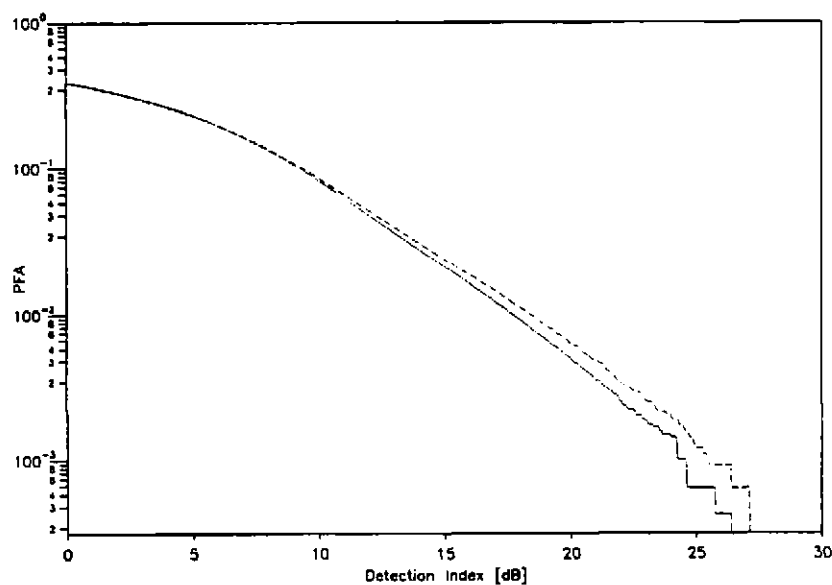


Figure 4.8: Probability of False Alarm (PFA) versus detection index (DI) for different bandwidths BW. Solid line: BW = 128 Hz, dashed line: BW = 32 Hz, dotted line: Rayleigh-approximation

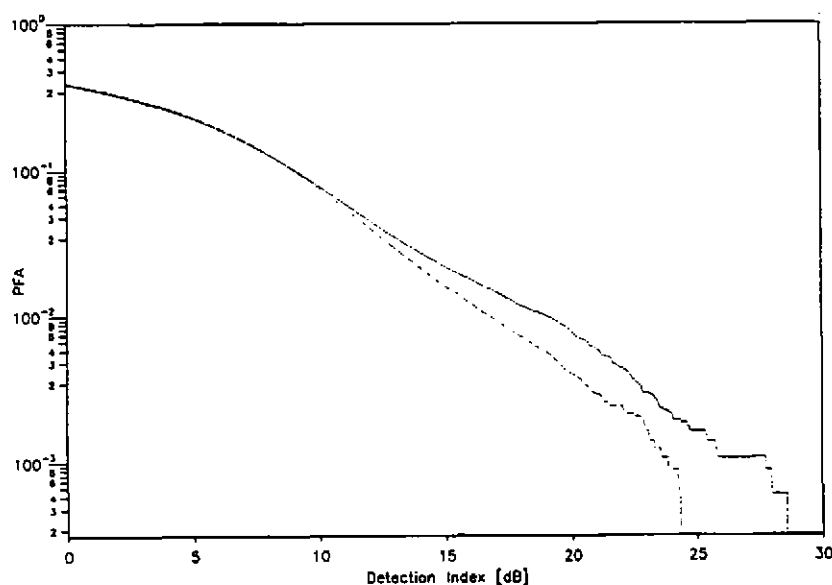


Figure 4.9: Probability of False Alarm (PFA) versus detection index (DI) for different directions. Solid line: transmission to starboard, dashed line: transm. to port, dotted line: Rayleigh-approximation
Proc. I.O.A. Vol.17 Part 8 (1995) 187

FALSE ALARM STATISTICS OF AREA-DEPENDENT REVERBERATION

5. CONCLUSIONS

The false alarm situation under reverberation limited conditions has been investigated, using data from an experimental towed array sonar system. The false alarm statistics was found to depend on parameters like location, azimuth angle, and sometimes bandwidth. The amplitude distribution function differs from known functions like Rayleigh or lognormal distribution and is highly area dependent. As a consequence, it cannot be described by only one or two parameters which are related to bottom type or propagation loss, for example. The whole distribution function or at least several of its central moments have to be known when the false alarm situation and thus the SONAR performance is to be estimated.

References

- [1] S. Stanic und E. Kennedy.
Reverberation fluctuations from a smooth seafloor.
IEEE Journal of Oceanic Engineering 18(2), 95-99, April 1993.
- [2] M. Gensane.
A Statistical study of acoustic signals backscattered from the sea bottom.
IEEE Journal of Oceanic Engineering 14(1), 84-93, January 1989.
- [3] J.-P. Babst T. M. Buzug und J. Ziegenbein.
Image processing for activated towed array sonar displays.
proceedings of the OCEAN 94 OSATES, Vol. II, Brest/France, p. II455-II460, 1994.
- [4] F. Greiner.
Intensity coding concepts for sonar images.
Proceedings of IOA International Conference on SONAR Signal Processing, Loughborough, 1995.