A REAL TIME FPGA IMPLEMENTATION OF A YAW STABILISED BEAMFORMER FOR SAS, SIDE-SCAN, AND FORWARD LOOK MINE-HUNTING SONARS

S Banks Bloomsbury DSP Ltd., London, UK
R Hollett NATO Undersea Research Centre, La Spezia, Italy
S Charles Bloomsbury DSP Ltd., London, UK
A Willcox Bloomsbury DSP Ltd., London, UK
A Bellettini NATO Undersea Research Centre, La Spezia, Italy

1 INTRODUCTION

This paper describes the design and implementation of a reconfigurable real time broadband sonar beamformer that can be used for a wide range of sonar applications. Applications include: Synthetic Aperture Sonar (SAS), side-scan sonar, and forward look mine hunting sonars.

The beamformer addresses the growing demand for compact and low power signal processing solutions for sonar systems integrated into autonomous underwater and autonomous surface vehicles. There is a growing interest in the use of such vehicles, fitted with high performance sonar systems, for mine counter measures operations and also in the offshore survey industry. The space, power and cooling constraints in such vehicles places new demands on the technology used for sonar signal processing.

Facility to compensate for platform yaw, which can be measured using an INS (Inertial Navigation System), is provided. The ability to process broadband data means the design can operate with many pulse functions, for example, chirped sonar data. By varying a set of coefficients, the beamformer can be reconfigured to produce polar beams spaced in angle, or parallel beams spaced along the direction of travel. This reconfigurability means the beamformer can be easily adapted for use in many different applications.

The implementation uses a combination of FPGA and GPP (General Purpose Processor) technology. The core algorithm is implemented using an FPGA, with control and data formatting functionality being carried out using a PowerPC processor. The implementation is low power and compact, making it suitable for use in applications such as AUVs[1]. Flexibility is retained through the ability to change beamforming coefficient sets, and the use of field programmable logic.

It is likely that the design could be extended to include stabilisation of sway and surge to form a real time fully stabilized backprojector.

2 BEAMFORMER ALGORITHM

For linear equi-spaced arrays beamforming can be efficiently carried out in the frequency domain. Such techniques are highly applicable to the *physical* aperture of synthetic aperture sonars, multi-beam side-scan sonars and forward look sonars. In the case of synthetic aperture beamforming, additional processing stages can be used for SAS stacking that account for non-linear, non equi-spaced *synthetic* apertures. This document concentrates primarily on the formation of physical aperture beams.

The algorithm proceeds in three stages:

1. **Beam stabilisation** corrects the sonar data for platform yaw and wavefront curvature.

2. **Frequency domain beamformer** forms beams using an azimuth FFT. Beam angles are frequency dependent and spaced in $\sin(\theta)$.
3. **Polar interpolator** creates beams spaced at user defined angles (other than θ) and removes frequency dependence on angle.

Figure 1 depicts a near-field arrival impinging on a moving linear array yawed at angle Φ with respect to its direction of motion. The wavefront curvature is shown as the blue dashed line. A local co-ordinate system (x, y) is used with the origin placed at the centre of the array. The x axis is along the direction of motion, the y axis broadside to the direction of motion. θ is the angle from broadside, +ve clockwise. The sonar data are processed in range cells of length n1 samples.

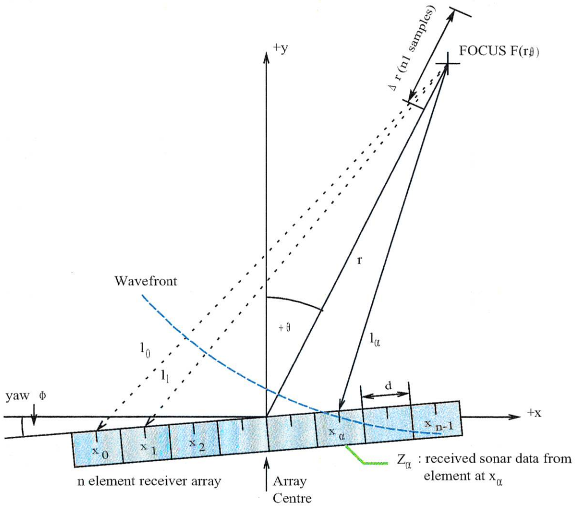

Figure 1: Near field arrival at receiver array from a sector about broadside

2.1.1 Stage 1 : Beam stabilisation

The path length to the sensor at x_α is given by:

$$l_\alpha = r\sqrt{(x_\alpha^2 / r^2 - 2x_\alpha / r \sin(\theta) + 1)} \qquad \text{...(1)}$$

where $\alpha = 0,...,n-1$ and the focal point *F* is specified in local polar coordinates (r,θ). The square root in **(1)** is approximated by Taylor expansion to second order:

$$l_\alpha \approx \tfrac{1}{2} x_\alpha^2 / r \cos^2(\theta) - x_\alpha \sin(\theta) + r \qquad \text{...(2)}$$

A frequency dependent phase shift with respect to the centre of the array (zero phase shift at the centre) can now be written:

$$\exp(j2\pi k_\gamma(l_\alpha - r)) \approx \exp(j2\pi k_\gamma(\tfrac{1}{2}x_\alpha^2 / r\cos^2(\theta) - x_\alpha \sin(\theta)))$$

$$k_\gamma = (f_c + f_\gamma)/c \ (\mathrm{m}^{-1})$$

...(3)

where f_c is the centre frequency (Hz) and f_γ (Hz) is a frequency vector over the baseband signal, $\gamma = -\frac{n1}{2},\ldots,\frac{n1}{2} - 1$ where n1 is the range cell length in samples. Noting that $\cos^2(\theta) \approx 1$ in a narrow sector about broadside:

$$\exp(j2\pi k_\gamma(l_\alpha - r)) \approx \exp(j2\pi k_\gamma x_\alpha^2 / 2r)\exp(-j2\pi k x_\alpha \sin(\theta))$$

...(4)

The received data $Z_\alpha(\theta)$ are multiplied by the quadratic range dependant phase shift $\exp(j2\pi k_\gamma x_\alpha^2 / 2r)$ from **(4)** along with a correction for platform yaw $\exp(-x_\alpha \sin(\Phi))$. The correction for yaw aligns the beams with the coordinate system (x, y). The frequency dependant phase shifts are applied using a range FFT over *n1* samples within the range cell Δr shown in Figure1. Hence the data corrected for wavefront curvature and platform yaw are:

$$Zcor_\alpha(\theta) = P_\alpha Z_\alpha(\theta)$$

...(5)

where

$$P_\alpha = \exp(j2\pi k_\gamma x_\alpha^2 / 2r)\exp(-x_\alpha \sin(\Phi))$$

...(6)

2.1.2 Stage 2 : Frequency domain beamformer

$Zcor_\alpha(\theta)$ is beamformed using an azimuth FFT (along the array) of length *n2* to form a set of beams $Z_\beta(\theta)$ where $\beta = -\frac{n2}{2},\ldots,\frac{n2}{2} - 1$. For an *n* element array with element spacing *d*, and forming *n2* beams:

$$Z_\beta(\theta') = \sum_{\alpha=0}^{n-1} \exp(j\frac{2\pi}{n2}\alpha\beta) Zcor_\alpha(\theta)$$

...(7)

Beam angles are dependant on sensor spacing *d* (see Figure 1), frequency, and transform size $n2 \geq n$. Beams are formed at a set of angles $\theta' = \sin^{-1}((2\pi / n2)(\beta / k_\gamma d))$. The resulting polar beams are shown in Figure 2.

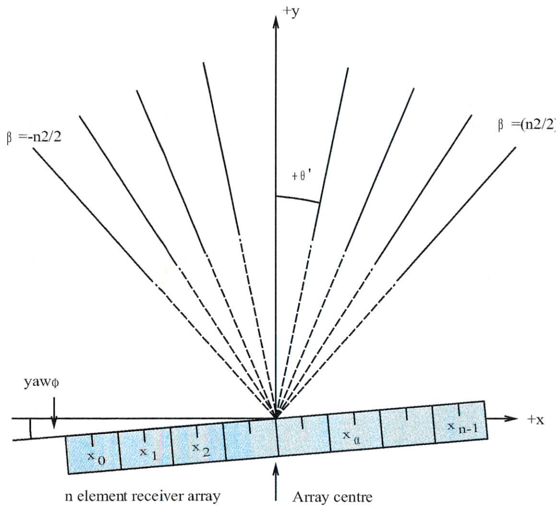

Figure 2: Polar beams formed by polar beamformer

2.1.3 Stage 3: Polar interpolator

The polar interpolator allows beams to be formed in directions θ'' other than the set θ' by interpolating in the spatial domain. The interpolator also removes the frequency dependence of the beam angle. The interpolator is derived by reformulating the expression for $Z_\beta(\theta')$ **(7)** in terms of spatial spectra.

$$Z(\theta'') = \frac{1}{n2^2}\sum_{\alpha=0}^{n-1}\sum_{\beta=-\frac{n2}{2}}^{\frac{n2}{2}-1}\sum_{\beta'=-\frac{n2}{2}}^{\frac{n2}{2}-1}\exp(-j\frac{2\pi}{n2}\alpha(\beta+\beta'))P_\beta Z_{\beta'}(\theta')$$

By excluding summations to zero, i.e. when $\beta \neq \beta'$, the expression can be reduced to:

$$Z(\theta'') = \frac{1}{n2}\sum_{\beta=-\frac{n2}{2}}^{\frac{n2}{2}-1} P_\beta Z_{-\beta}(\theta') \qquad \text{...(8)}$$

where

$$P_\beta = \exp(-j\pi(n-1)\frac{\beta'}{n2})\sum_{\alpha=0}^{n-1}\exp(j\frac{2\pi}{n2}\alpha(\beta+\beta')),\ \beta' = \frac{n2}{2\pi}k_\gamma d\sin(\theta'')$$

In principle, the interpolator P_β is implemented over the full spectral range $(-n2/2 \le \beta \le n2/2-1)$. In practice, P_β is truncated to $(-L \le [\beta'] \le L)$ about the nearest integer to β' $([\beta'])$, depending on the accuracy required. For speed, the coefficients are stored offline for subsequent retrieval.

The polar grid is implemented by interpolating the beamformer output in directions equispaced in azimuth θ''; the trapezoidal (rectangular), grid in directions equispaced in $\tan(\theta'')$. n3 rectangular beams formed by the interpolator are shown in Figure 3.

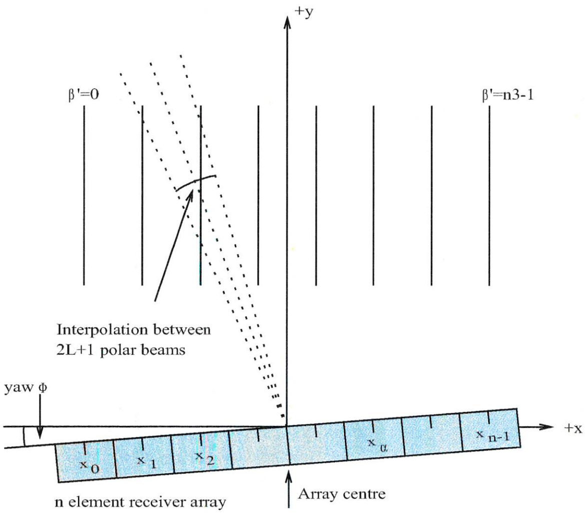

Figure 3: Beams output from polar interpolator – example shows parallel beams

2.1.4 Depth of field in range cell processing

In principle, the correction for wavefront curvature in **(5)** is valid at only a single range point r. In practice, a fixed correction is applied over a range interval or range cell. Typically the cell size is chosen so that the amount of defocusing at the cell edges is as large as can be tolerated. The size of the cell can then be regarded as a depth of field. The criterion adopted for depth of field is that the wavefront deviate by no more than $\pm\lambda/8$ over the array. This leads to a depth of field D given by:

$$D = 2\lambda\left(\frac{r}{nd}\right)^2 \qquad \text{...(9)}$$

for an array of length nd and signal (carrier) wavelength λ.

2.2 Implementation of the algorithm

The algorithm has been implemented using the Xilinx FPGA devices provided on the Spectrum Signal Processing SDR 3000 system[2]. Some control functions are implemented on a PowerPC 7410 processor, also provided by the Spectrum 3000 system. The processing was designed to operate with various data sets. Output from the beamformer is in raw or Triton XTF format.

The SDR 3000 system consists of a Pro-3500 GPP board and a Pro-3100 FPGA board. The GPP board provides two PowerPC 7410 processors, with Altivec extensions, operating at 500MHz. The FPGA board provides four Xilinx Virtex II 6 mega-gate FPGAs.

The design is able to process 100 sonar data channels, sampled at 75kHz (complex baseband) in real time. INS sensor data can be used to provide a real time update of platform yaw, which can be fed into the beamformer to produce a yaw stabilised output.

The algorithm divides well into the following two sections :

1. **Systematic processing** including FFT, vector phase multiplications, beamforming and inverse FFT.

2. **Non-systematic processing** including control of data loading and unloading, generation of coefficient sets (which are subsequently stored as a look-up table in RAM) and formatting of data in Triton XTF format.

Systematic algorithms are well suited to FPGA implementation. Configured for a specific task, FPGAs can provide significantly higher processing power per watt and reduced physical volume, compared to a GPP. For this reason, the FFT, cross correlation and beamforming were implemented using a FPGA device.

GPP devices, such as the PowerPC provided on the Spectrum SDR 3000 system, are well suited to non-systematic algorithms such as control. Coding using a language such as C means changes to the algorithm are relatively easy to make. For this reason, control of the beamforming process was implemented in a PowerPC processor. The PowerPC processor also provides debugging functionality.

2.3 FPGA Implementation

The original floating point reference implementation was converted to use fixed point (integer) data types. Sonar data are stored in complex baseband format (16 bit real, plus 16 bit imaginary). Phase coefficients are generally represented as (8 bit real, 8 bit imaginary) complex data.

Data processing is carried out using range cells fixed to a length of n1=512 samples. 512 was chosen as a balance between computational accuracy (i.e., minimum defocusing in the range cell), and good FPGA utilisation. A common matrix memory with facility to stream data in range or columns is used for all stages described in section 2. Two separate computational blocks are used; these are described in the following sections.

2.3.1 Stabilisation and beamforming (stages 1 and 2)

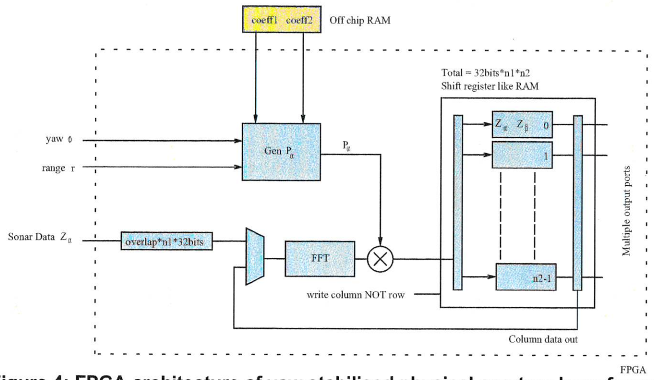

Figure 4: FPGA architecture of yaw stabilised physical aperture beamformer

The physical aperture beamformer architecture is shown in Figure 4. The design uses one streaming FFT unit, capable of performing forward and inverse FFTs on data up to 512 samples in length. A complex multiplier is used to apply the beam stabilisation coefficients P_α (equation **(6)**). Adjustment of these coefficients to compensate for platform yaw (equation **(6)**), is achieved using a sin/cos lookup table inside the block 'Gen P_α '.

The current range cell is stored in the shift register like RAM. The first two stages, described in **(2)**, are carried out as follows:

1. A range FFT, length $n1 = 512$, transforms the pulse compressed sonar data into the frequency domain. The resulting data are then multiplied by the phase factor P_α to correct for both yaw and the curvature of the arriving wavefronts, before being written into the matrix RAM structure row by row.

2. A Fourier transform, length $n2 = 128$ is applied in azimuth to give the wavenumber data. For this stage, data are streamed out of the RAM column by column, through the FFT unit, and then back into the matrix RAM.

The resulting beams are in the temporal frequency domain; the final inverse FFT is performed in the polar interpolation stage (described in the next section).

2.3.2 Polar interpolation

The architecture to implement the polar interpolation is shown in Figure 5. The matrix RAM, used in the previous stage, is repeated for clarity.

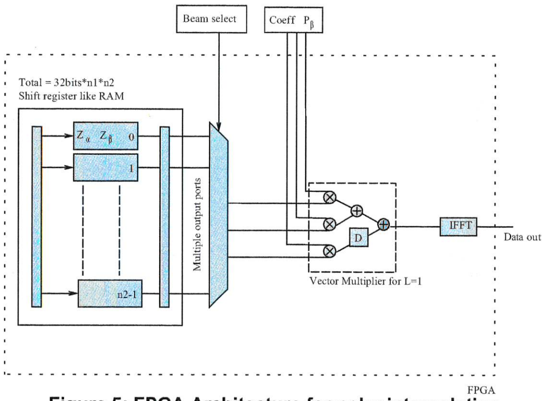

Figure 5: FPGA Architecture for polar interpolation

The right hand side of the RAM block facilitates reading from up to three memories simultaneously. The high memory data rate this achieves means the polar interpolator can stream data out (one sample per clock cycle). Three complex multiplications along with the vector sum are also performed on each clock cycle. The data are streamed through an inverse FFT and then gated to account for a specified number of overlapping samples between range cells.

Coefficient data are stored in off chip RAM. These data are relatively large as values must be stored for each frequency bin, for each polar beam, for each possible value of *n* where *n* is the maximum number of sonar receiver channels. To enable the architecture to stream, it is necessary to read three complex values from RAM per clock cycle. This assumes L, the length parameter of the polar interpolator, is set to a value of 1 (i.e., effect length 2L+1 = 3). Because the memory on the Spectrum 3100 boards is limited to 64 bit data bus, it is necessary to pack these coefficients into 3*(9 bits real + 9 bits imaginary) = 54 bits.

A compacted version of the beam select table is stored in off-chip memory. The compaction limits the memory to storing only the centre beam; it is assumed that the adjacent beams will always be either side of the centre beam. The values read from the beam select table are used to select the appropriate three memories using the multiplexer shown in Figure 5.

2.4 Power PC Implementation

The loading and unloading of data, along with coefficient generation, is controlled by a PowerPC processor. All processing of sonar data is carried out using the FPGA. The PowerPC also provides data formatting options, such as the ability to output data in XTF format.

3 APPLICATIONS OF THE BEAMFORMER

3.1 Synthetic Aperture Sonar

The beamformer is suitable for use as a physical aperture beamformer in Synthetic Aperture Sonar (SAS) processing systems. Such systems would contain the additional components shown in Figure 6.

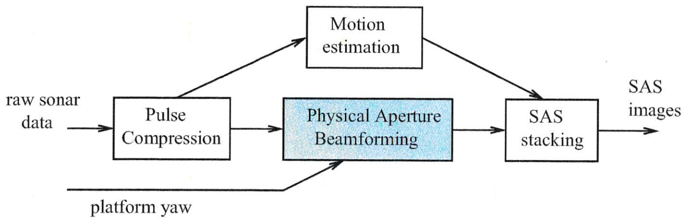

Figure 6: Processing stages for a Synthetic Aperture Sonar

3.2 Forward look sonar

Forward look sonars are commonly used in Mine Counter Measures (MCM) operations. The ability to correct for platform yaw in real time means that the beamformer can be used to create a yaw stabilised image. The ability to automatically stabilise the image makes operating the sonar easier.

Figure 7 shows the effect of yaw stabilisation on a forward look sonar producing beams spaced in angle. The beamformer is also capable of producing parallel beams, which can also be used for forward look sonar data.

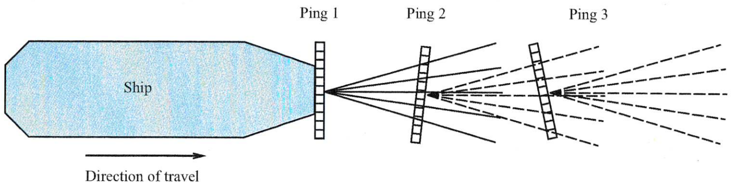

Figure 7: Yaw stabilised forward look sonar

3.3 Multibeam sidescan

Multibeam sidescan allows the platform tow speed to be increased compared to standard single beam sidescan. This is because several sonar beams at different physical locations can be formed from each ping. 'Black spots' between sonar pings at close range can also be reduced with multibeam sonar. The ability to compensate for platform yaw in real time significantly reduces the amount of post-processing required to produce a geo-referenced sidescan image.

When used in a multibeam sidescan system, the interpolation coefficients are set such that the beams are output in parallel, suitable for placing directly onto the final sonar image.

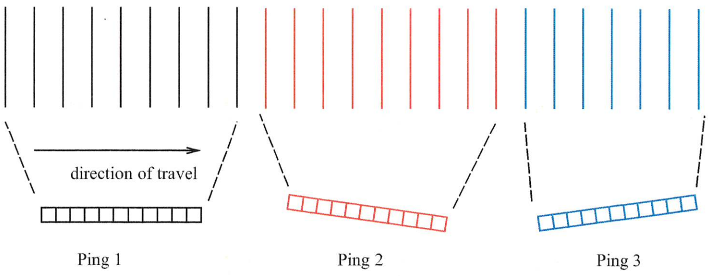

Figure 8: Sonar image formed from multibeam sonar

4 MODIFICATIONS TO FORM A FULLY STABILIZED BACKPROJECTOR

The beamformer presented here corrects, in real time, for platform yaw, producing yaw stabilised focused beams. The design could also be modified to correct for platform sway and surge, producing a fully stabilised backprojector. With such a backprojector, the extension to SAS imaging is a simple coherent summation of the focused physical aperture images. This is because a stabilised backprojector is able to output physical aperture images that lie exactly on top of each other in space.

A fully stabilised backprojector is also applicable to multibeam sidescan, replacing most of the geo-referencing post processing that takes place in a traditional sidescan sonar. This could potentially result in fully geo-referenced images being produced in real time straight from the sonar system.

5 RESULTS

Figure 9 shows the results of processing a point scatterer, at 75m, using the beamforming method proposed in this paper. The array is yawed at 2 degrees in the simulation. (The array is based on a state-of-the-art SAS design for optimal area coverage, operating at 300 kHz with 60 kHz band and comprising 36 sensors equispaced over an aperture of 1.2 m.)

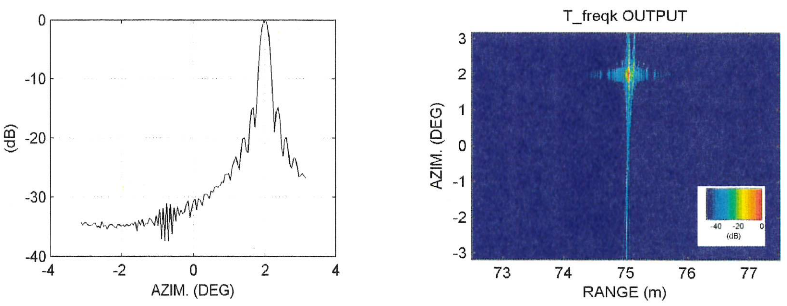

Figure 9: Results using a simulated point scatterer at 75m

6 CONCLUSIONS

This paper has presented a novel algorithm for fully focused, yaw stabilised sonar beamforming. The implementation, using FPGA devices, leads to a low power and compact solution that can be embedded into sonar systems. The ability to change interpolation coefficients means that the design is highly flexible.

The beamformer presented has a number of sonar applications including physical aperture beamforming for Synthetic Aperture Sonar, forward look sonar and also multibeam sidescan sonar.

The beamformer can easily be adapted to compensate for sway and surge motion errors, creating a fully stabilised backprojector.

7 ACKNOWLEDGEMENTS

Peter Munk, Marc Pinto and Ben Evans from the NATO Undersea Research Centre, La Spezia, Italy, are gratefully acknowledged for their contributions to this work.

8 REFERENCES

1. S. Banks, S. Charles, A. Willcox, 'FPGA Based real time synthetic aperture sonar processing for AUVs', IOA conference on Synthetic Aperture Sonar and Synthetic Aperture Radar, Lerici, Italy, (September 2006).
2. Spectrum Signal Processing, see: http://www.spectrumsignal.com/*

*Addresses correct at time of writing