

Implementation of Adaptive Digital Beamforming using Cordic

AZRA JEELANI
Associate Professor,
M S Engineering College,
Bangalore, Karnataka, India
azrajeelani@gmail.com

Dr. VEENA.M.B
Associate Professor,
B M S College of Engineering,
Bangalore, Karnataka, India.
veenamb.ece@bmsce.com

[Dr. CYRIL](#) PRASANNA RAJ
Dean R & D,
M S Engineering College,
Bangalore, Karnataka, India
cyril@msec.ac.in

Abstract: Sonar imaging is one of the simplest technique for detection of under water drowned bodies. There is a need for design of conventional beamforming which are robust and simple. Adaptive beamformer is used to improve the quality of the sonar image. As a result we get an image containing more useful and correct information. The CORDIC computing technique a highly efficient method to compute elementary functions like sine, cosine, translate, rotate values using CORDIC algorithm. The system simulation was carried out using ModelSim and Xilinx ISE Design Suite 9.2i.. Matlab code is used to implement sin and cos using cordic angles and amplitude response of beamformed data by optimized method in order to enlarge the validity region of beamforming. Synthesis results of cordic shows the reduced memory requirement and less power consumption.

Keywords: Beamforming , cordic, sonar imaging, validity region

1. INTRODUCTION

Beamforming is a type of signal processing technique used in sensor arrays for directional signal transmission or reception. Here the elements are combined in such a way that signals at particular angles experience constructive interference while others experience destructive interference. 3D sonar imaging has been one of the main innovations in underwater applications over the years[1]. There are two critical issues in the development of high resolution 3D sonar systems are 1) the cost of hardware, which is associated with the huge number of sensors that compose the planar array and 2) the computational burden in processing the signals. Palmese and Trucco also propose an algorithm to perform chirp zeta transform beam forming on the wideband signals collected by an evenly spaced planar array and generated by a scene placed in both the far field and the near field [4],[6]. Works are done in [8]-[10] have proposed to use the Coordinated Rotation Digital Computer(CORDIC) in implementing frequency domain beamforming on field Programmable Gate Arrays-the CORDIC algorithm in an iterative arithmetic algorithm given by Volder[11] and Walther[12].This paper describes a data path using CORDIC for the algorithm.

The digital signal processing has long been dominated by microprocessors with enhancements such as single cycle multiply-accumulate instructions and special addressing modes. While these processors are low cost and offer extreme flexibility, they are not fast enough for truly demanding DSP tasks. The advent of high speeds of dedicated hardware solutions which has the costs that are competitive with the traditional software approach. Unfortunately, algorithms optimized for these microprocessor based systems do not always map well into hardware. While hardware-efficient solutions often exist, the dominance of the software systems has kept those solutions out of the spotlight. Among these hardware-efficient algorithms is a class of iterative solutions for trigonometric and other functions that use only shifts and adds to perform. The trigonometric functions are based on vector rotations, while other functions like square root are implemented using an incremental expression of the desired function. The trigonometric algorithm is called CORDIC, an acronym for COordinate Rotation DIgital Computer. The

incremental functions are performed with a very simple extension to the hardware architecture, and while not CORDIC in the strict sense, are often included because of the close similarity. The CORDIC algorithms generally produce one additional bit of accuracy for each iteration. The trigonometric CORDIC algorithms were originally developed as a digital solution for real-time navigation problems. The original work is credited to Jack Volder [4,9]. Extensions to the CORDIC theory based on work by John Walther[1] and others provide solutions to a broader class of functions. This paper attempts to survey the existing CORDIC and CORDIC-like algorithms and then towards implementation in Field Programmable Gate Arrays (FPGAs).

A approximation of used in near field beamforming presented in[13],[14] by enlarging the validity region. Beamforming can be used at both the transmitting and receiving ends in order to achieve spatial selectivity. To change the directionality of the array when transmitting, a beamformer controls the phase and relative amplitude of the signal at each transmitter, in order to create a pattern of constructive and destructive interference in the wave front. When receiving, information from different sensors is combined in a way where the expected pattern of radiation is preferentially observed. Conventional beamformers use a fixed set of weightings and time-delays (or phasings) to combine the signals from the sensors in the array, primarily using only information about the location of the sensors in space and the wave directions of interest. In contrast, adaptive beamforming techniques generally combine this information with properties of the signals actually received by the array, typically to improve rejection of unwanted signals from other directions. This process may be carried out in either the time or the frequency domain. Hardware implementation of bio-inspired algorithm for motion detection takes less processing time. Integration of motion detection model and improves the performance of autonomous visual navigation. For resolving the navigation problems two existing approach optical flow or non bio-inspired and bio- inspired processing time is needed to reduce. For minimizing the size of system algorithm should be implemented on ASIC and functionality should be verified on

FPGA before taking to ASIC.

2. BACKGROUND THEORY:

2.1. Beamforming:

Beamforming is a type of signal processing technique used in sensor arrays for directional signal transmission or reception. Here the elements are combined in such a way that signals at particular angles experience constructive interference while others experience destructive interference[1]. Beamformers are classified as either data independent or statistically optimum, depending on how the weights are chosen. The weights in a data independent beamformer do not depend on the array data and are chosen to present a specified response for all signal and interference scenarios. The weights in a statistically optimum beamformer are chosen based on the statistics of the array data to optimize the array response. The statistics of the array data are not usually known and may change over time so adaptive algorithms are typically used to determine the weights. The adaptive algorithm is designed so the beamformer response converges to a statistically optimum solution [6].

The weights in a data independent beam former are designed so that the beamformer response approximates a desired response independent of the array data or data statistics. This design objective is same as that for a classical FIR filter design. The simple delay and sum beam former is an example of the data independent beamforming.

In statistically optimum beam former the weights are chosen based on the statistics of the data received at the array. The goal is to optimize the beam former response so that the output signal contains minimal contributions due to the noise and signals arriving from directions other than the desired direction. The Frost beamformer is a statistically optimum beam former. Other statistically optimum beamformers are Multiple Side lobe Canceller and Maximization of the signal to noise ratio.

2.2. Sonar Imaging:

Sonar (an acronym for SOund Navigation and Ranging) is a technique that uses sound propagation (usually underwater, as in submarine navigation) to navigate, communicate with or detect objects on or under the surface of the water, such as other vessels. Two types of technology share the name "sonar": passive sonar is essentially listening for the sound made by vessels; active sonar is emitting pulses of sounds and listening for echoes. Sonar may be used as a means of acoustic location and of measurement of the echo characteristics of "targets" in the water. Acoustic location in air was used before the introduction of radar. Sonar may also be used in air for robot navigation, and SODAR (upward looking in-air sonar) is used for atmospheric investigations. The term sonar is also used for the equipment used to generate and receive the sound.

2.3. Active and Passive Sonar System

Active sonar or passive sonar, when receiving the acoustic signal reflected from the target, the information included in the signal cannot be directly collected and used without technical signal processing. To extract the efficient and useful information's from the mixed signal, some steps should be taken to transfer sonar data from raw acoustic data reception to detection output.

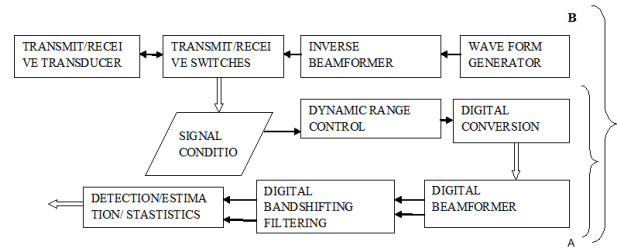


Fig.1 Passive and Active Sonar System

$A = \pi r^2$ needed during the signal processing system as shown in Fig.1

2.4. Cordic Theory:

Coordinate Rotational Digital Computer (CORDIC) is a set of shift-add algorithm known for computing a wide range trigonometric functions, hyperbolic, linear and logarithmic functions also like multiplication division, data type conversion, square root. It is highly efficient, low complexity. The CORDIC algorithm has found in various applications such as pocket calculator, numerical co-processors to high performers Radar signal processing, supersonic bomber. Vector rotation can also be used for polar to rectangular and rectangular to polar conversions, for vector magnitude, and as a building block in certain transforms such as the DFT and DCT. The CORDIC algorithm provides an iterative method of performing vector rotations by arbitrary angles using only shifts and adds. The algorithm, credited to Volder[4], is derived from the general (Givens) rotation transform:

$$x' = x \cdot \cos(\phi) - y \cdot \sin(\phi) \quad \text{-----(1)}$$

$$y' = x \cdot \sin(\phi) + y \cdot \cos(\phi) \quad \text{-----(2)}$$

which rotates a vector in a Cartesian plane by the angle ϕ . These can be rearranged so that:

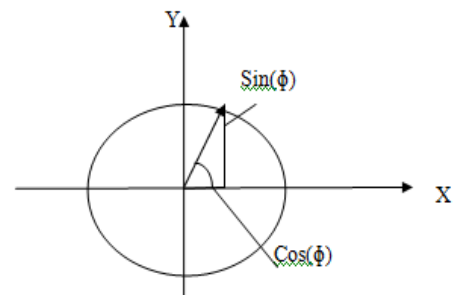


Fig 2. Rotation of sin and cos

$$x' = \cos(\phi) \cdot [x - y \cdot \tan(\phi)] \quad \text{-----(3)}$$

$$y' = \cos(\phi) \cdot [x + y \cdot \tan(\phi)] \quad \text{-----(4)}$$

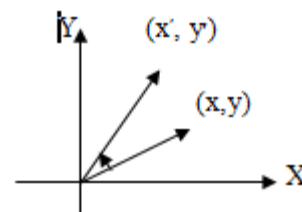


Fig 3. Input and output of rotation for rotation mode

So far, nothing is simplified. However, if the rotation angles are restricted so that $\tan(\phi)2^{-i}$, the multiplication by the tangent term is reduced to simple shift operation. Arbitrary angles of rotation are obtainable by performing a series of successively smaller elementary rotations. If the decision at each iteration, i , is which direction to rotate rather than whether or not to rotate, then the $\cos(\delta_i)$ term becomes a constant (because $\cos(\delta_i) = \cos(-\delta_i)$). The iterative rotation can now be expressed as:

$$X_{i+1} = K_i [x_i - d_i y_i 2^{-i}] \quad \text{-----(5)}$$

$$Y_{i+1} = K_i [x_i + d_i y_i 2^{-i}] \quad \text{-----(6)}$$

Where

$$K_i = \cos(\tan^{-1} 2^{-i}) = 1/\sqrt{1+2^{-2i}} \quad \text{----(7)}$$

$$d_i = \pm 1 \quad \text{-----(8)}$$

Removing the scale constant from the iterative equations yields a shift-add algorithm or vector rotation. The product of the K_i 's can be applied elsewhere in the system or treated as part of a system processing gain. That product approaches 0.6073 as the number of iterations goes to i infinity. Therefore, the rotation algorithm has a gain, A_n of approximately 1.647. The exact gain depends on the number of iterations, and obeys the relation

$$A_n = \prod \sqrt{1+2^{-2i}} \quad \text{-----(9)}$$

The angle of a composite rotation is uniquely defined by the sequence of the directions of the elementary rotations. That sequence can be represented by a decision vector. The set of all possible decision vectors is an angular measurement system based on binary arctangents. Conversions between this angular system and any other can be accomplished using look-up. A better conversion method uses an additional adder-subtractor that accumulates the elementary rotation angles at each iteration. The elementary angles can be expressed in any convenient angular unit. Those angular values are supplied by a small lookup table (one entry per iteration) or are hardwired, depending on the implementation. The angle accumulator adds a third difference equation to the algorithm.

$$Z_{i+1} = Z_i + d_i (\tan^{-1} 2^{-i}) \quad \text{-----(10)}$$

Obviously, in cases where the angle is useful in the arctangent base, this extra element is not needed. The CORDIC rotator is normally operated in one of two modes. The first, called rotation by Volder[4], rotates the input vector by a specified angle (given as an argument). The second mode, called vectoring, rotates the input vector to the x axis

2.5. Implementation in an FPGA

There are a number of ways to implement a CORDIC processor. The ideal architecture depends on the speed versus area tradeoffs in the intended application. First we will examine an iterative architecture that is a direct translation from the CORDIC equations. From there, we will look at a minimum hardware solution and a maximum performance solution.

2.6. Iterative CORDIC Processors

An iterative CORDIC architecture can be obtained simply by

duplicating each of the three difference equations in hardware as shown in Figure 1. The decision function, d_i , is driven by the sign of the y or z register depending on whether it is operated in rotation or vectoring mode. In operation, the initial values are loaded via multiplexers into the x , y and z registers. Then on each of the next n clock cycles, the values from the registers are passed through the shifters and adder-subtractors and the results placed back in the registers. The shifters are modified on each iteration to cause the desired shift for the iteration. Likewise, the ROM address is incremented on each iteration so that the appropriate elementary angle value is presented to the z adder-subtractor. On the last iteration, the results are read directly from the adder-subtractors. Obviously, a simple state machine is required keep track of the current iteration, and to select the degree of shift and ROM address for each iteration. The design depicted in Figure 1 uses word-wide data paths (called bit-parallel design). The bit-parallel variable shift shifters do not map well to FPGA architectures because of the high fan-in required. If implemented, those shifters will typically require several layers of logic (i.e., the signal will need to pass through a number of FPGA cells). The result is a slow design that uses a large number of logic cells.

3. PROPOSED WORK:

Digital input pulse is passed to find the angle or detection of object under water. In Fig.3. Beamforming can be used at both the transmitting and receiving ends in order to achieve spatial selectivity, the data is transmitted to underwater sonar system, sonar is used to detect the underwater objects and finds the angle elevation. The beamformed data is transmitted; at the receiver end beam formation data is generated. The generated beam formation data will be having interference and noise error that will be reduced by using optimization technique. Optimized cordic beamforming will eliminate all the interference which generated at receiver end. Final optimization beamforming data is obtained

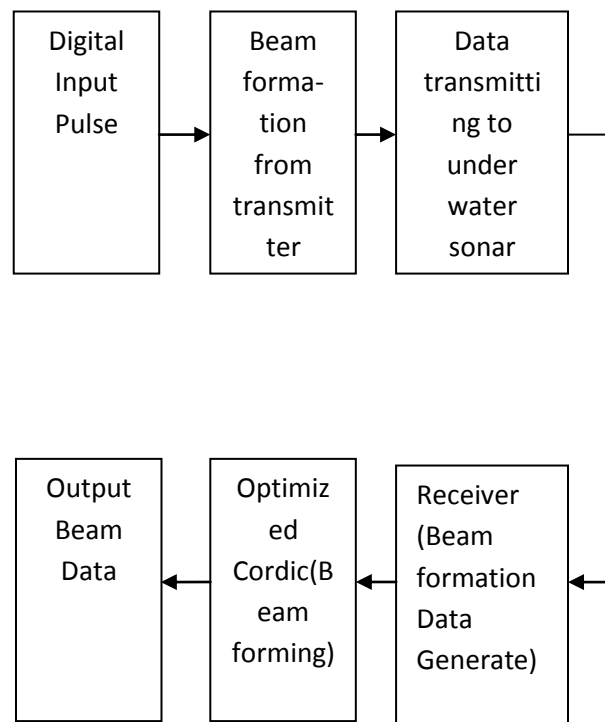


Fig 4.. Beamforming for underwater sonar

3.1. Program Flow Chart:

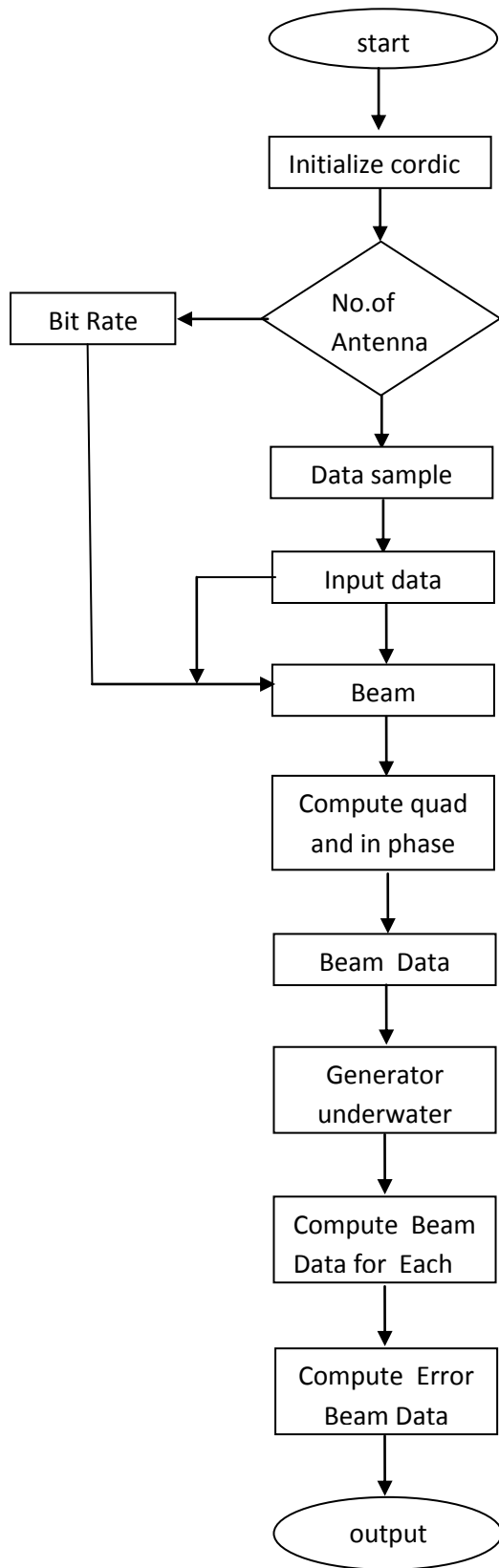


Fig.5 .Program Flow Chart

Fig 5. shows the flow chart in which initially the CORDIC values are sampled, the antennas are used to detect the angles and object of beam form underwater sonar with sampled bit rate. The detected angles are taken as input data where the beam data is formed, the obtained beam data are sampled according to mathematical calculations under CORDIC algorithm, the obtained beam data samples are computed as quad phase and In phase. The received beam data contains noise and interference which are reduced and eliminated using underwater noise model. The beam data is computed for each antenna and its angles, the error beam data is finally computed to obtain noiseless beam data. The obtained output is in the form of optimized beam form data.

3.2. Architecture:

The architecture is shown in Fig.6 in which input signals are given to memory. The memory is used to store the data of input signals. The signals are transmitted to detect the target or object in underwater beam form data. Once the target is detected and beam form data is generated. The received beam form data is up sampled and Down sampled. The adder is used to combine the images received and stored in memory. The generated beam form data signals from sonar are given to CORDIC algorithm

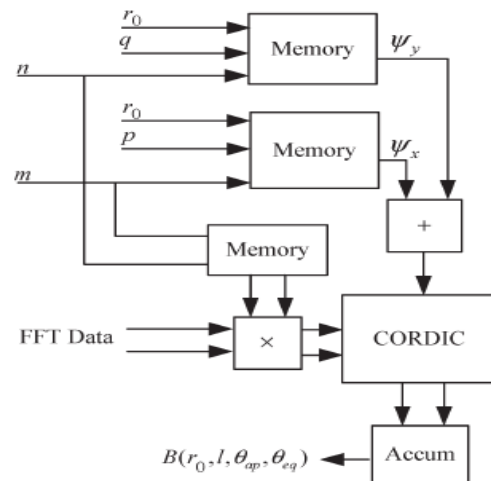


Fig.6. Data path algorithm

The received data is sampled according to CORDIC algorithm calculations. The Angle is measured using CORDIC. The sin and cos angles are generated and calculated using CORDIC algorithm. Both the IN Phase and Quad Phase is added and given to Cordic using adder. The CORDIC performs vector rotation and the vector data are some to produce the array beam (B). Angles which are detected are to be measured using CORDIC. The obtained Samples are stored in the register.

Rotation vector is given by equation (3 and (4)

To find iterations the following equations are used

$$X_{i+1} \alpha_i = x_i - d_i y_i 2^{-i} \quad \text{-----(11)}$$

$$Y_{i+1} \alpha_i = x_i + d_i y_i 2^{-i} \quad \text{-----(12)}$$

$$Z_{i+1} \alpha_i = x_i - d_i \arctan(2^{-i}) \quad \text{-----(13)}$$

To find magnitude and phase the following equations are used

$$X' = Z_n \sqrt{X^2 + Y^2} \quad \text{-----(14)}$$

$$Y' = 0 \quad \text{-----(15)}$$

$$\Theta' = \text{atan}(x/y) \quad \text{-----(16)}$$

Advantages and Disadvantages of CORDIC

- Simple Shift-and-add Operation.(2 adders+2 shifters vs. 4 mul.+2 adder)
- It needs n iterations to obtain n -bit precision.
- Slow carry-propagate addition.
- Low throughput rate and area consuming shifting operations.

The m and n are the input coordinates, p and q are pre-computed values, r_o are the rotation value, ψ_x and ψ_y is the phase shifter. By using L point DFT the sample data are calculated by

$$S_{m,n}(l) = \sum_{t=0}^{L-1} S_{m,n}(t) \exp(-j*2*\pi * t*l/L) ,$$

The $S_{m,n}(l)$ are stored in memory with indexing parameter m and n . The phase shift parameter ψ_x and ψ_y is added to the phase term of the data $W_{m,n} S_{m,n}(l)$ by CORDIC which perform as a vector rotation. The vector data is summed to produce the array beam of $B(r_o, \Theta_{ap}, \Theta_{eq})$ where l is the frequency, Θ_{ap} and Θ_{eq} is the time delay.

4. RESULTS & DISCUSSIONS:

4.1 Results of Direct Method Using MATLAB

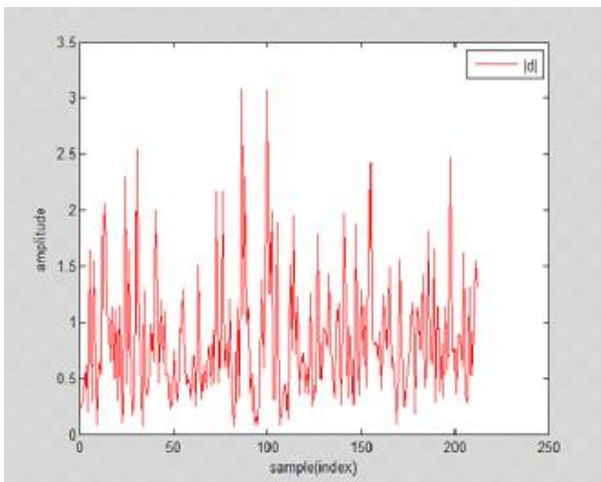


Fig.7 Amplitude of Transmitted Data

Fig.7 shows the variation based on the size of the data from the transmitter side. Fig.8 shows the phase wise changes from -10 degree to -40 degree based on the optimized algorithm.

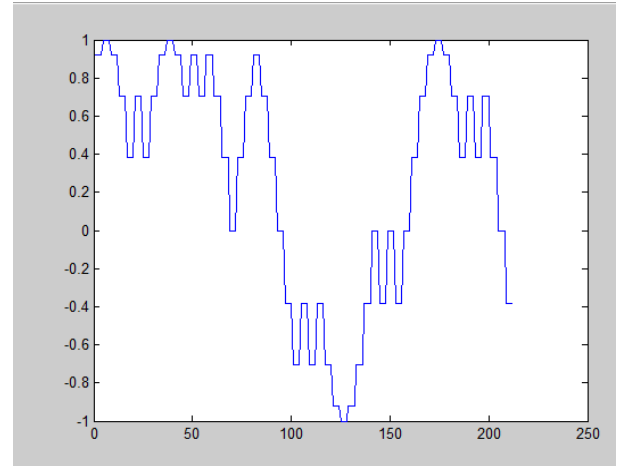


Fig.8. Input Data

In Fig.8 the Phase Graph of Input Data is transmitted.

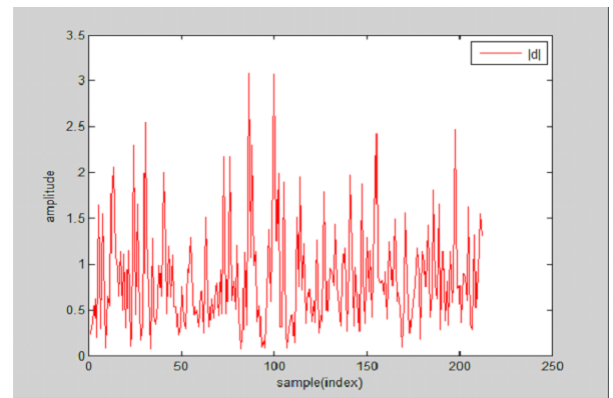


Fig.9 Amplitude of Transmitted data.

As shown in Fig 9. the amplitude variation based on the size of the data from the transmitter side and the amplitude variation generate from the beamforming data.

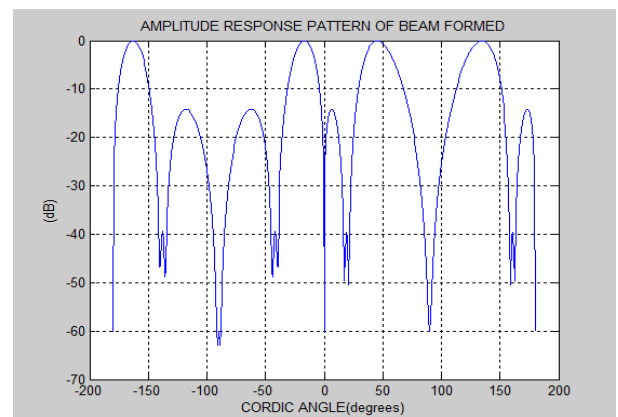


Fig.10 Output Data

As shown in fig 10 the Amplitude Response of Beamformed Data based on Cordic Angles, output is formed..

4.2. Results of Optimized Method Using MATLAB

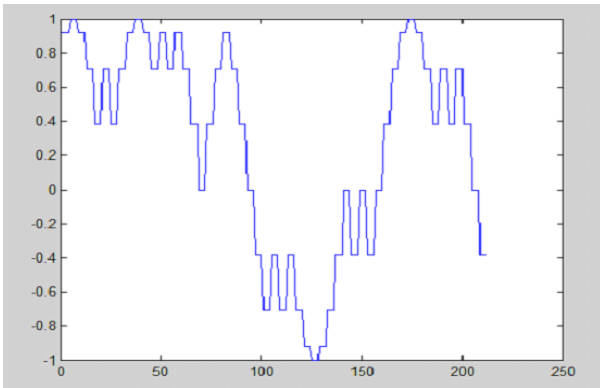


Fig.11. Phase Graph of Input Data

Fig.11. shows the input pulse sent from the transmitter side to underwater to detect the target.

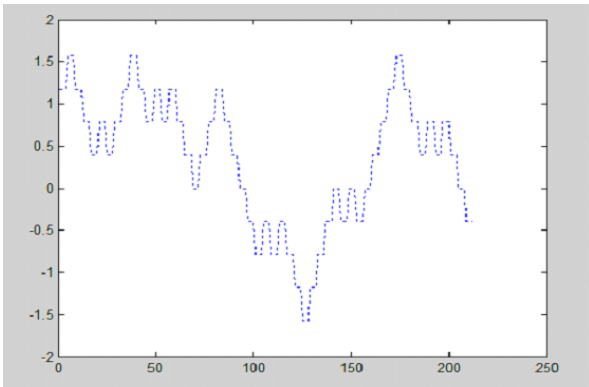


Fig.12. Beamformed Data.

Fig.12. shows the input data is sent in the form of samples from the transmitter side.

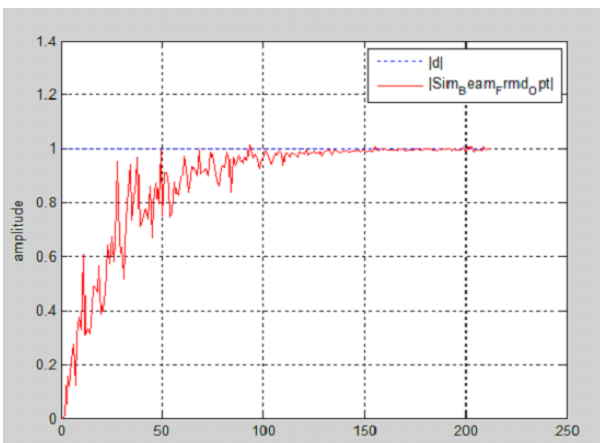


Fig.13. Amplitude Response Data

Fig.13. shows the amplitude response of optimized data from the transmitter to receiver.

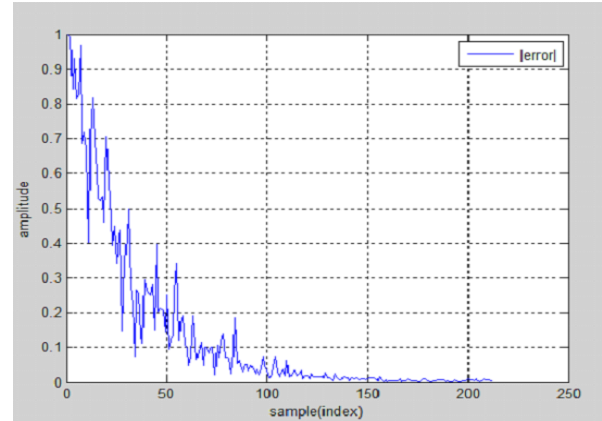


Fig.14. Error In beamformed Transmitted Data

Fig.14. shows the loss of data which is less compared to direct method.

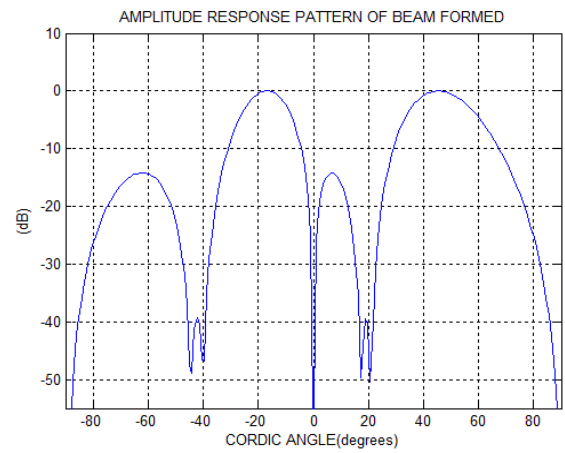


Fig.15. Amplitude Response of Beamformed Data Based on Cordic Angles

Fig.15. shows the amplitude response of beamformed data. At the transmitter side, the signal is up sampled and at the receiver side the signal is down sampled using cordic algorithm to get accurate result.

5. COMPARISON OF OPTIMIZED METHOD AND DIRECT METHOD:

Table 1. Comparison of direct method and optimized method

Parameter	Direct Method	Optimized Method	Memory reduced Optimized Method
Number of Delays per focusing distance	10^7 bytes	10^3 bytes	59kB
Validity range.	Does not enlarge.	Enlarged by 4 degree in azimuth and elevation angle	Enlarged by 4 degree
Computational requirement	More number of sensors	Less number of sensors.	Reduced by a factor of 2.

6. CONCLUSION:

This paper has illustrated that the proposed approximation enlarges the validity region of the system's view scene. Under the preferred definition of steering direction condition, the validity region is enlarged at least by 4° in both azimuth and elevation angles. The optimized algorithm has the advantage of reducing the memory and computational requirements as compared with DM beamforming. In high-resolution sonar systems, where more than ten thousands of beams are produced, the required memory for parameter storage is reduced.

Digital antennas have the potential of satisfying the requirements of many systems simultaneously. They are flexible, and capable of handling wide bandwidths, and can perform multiple functions. The bandwidth of the modulator and demodulator must match the bandwidth of the signal for efficient operation. The effects of the phase slope and amplitude variations on the pattern of a linear array were determined by simulations that incorporated the measured data. The simulation showed unacceptable beam squint with frequency.

7. REFERENCES:

- [1] V. Murino and A. Trucco, "Three-dimensional image generation and processing in under acoustic vision," vol. 88, no. 12, Dec 2000.
- [2] A. Davis and A. Lugsdin, "High speed underwater inspection for port and harbour security using coda Echoscope 3D sonar," 2005, pp. 2006-2011.
- [3] R. K. Hansen and P. A. Andersen, "The application of real time 3D acoustical imaging," OCEANS 1998, pp. 738-741.
- [4] M. Palmese and A. Trucco, "Digital near field beamforming for efficient 3-D underwater acoustic image generation," in Proc. IEEE Int. Workshop Imaging Syst. Tech., 2007, pp. 1-5.
- [5] M. Palmese and A. Trucco, "From 3-D sonar images to augmented reality models for objects buried on the seafloor," IEEE Trans. Instrum. Meas., vol. 57, no. 4, pp. 820-828, Apr. 2008.
- [6] M. Palmese, G. De Toni, and A. Trucco, "3-D underwater acoustic imaging by an efficient frequency domain beamforming," in Proc. IEEE Int. Workshop Imaging Syst. Tech., 2006, pp. 86-90.
- [7] B. E. Nelson, "Configurable computing and sonar processing-architecture and implementations", 2001, pp. 56-60.
- [8] B. L. Hutchings and B. E. Nelson, "Gigaop DSP on FPGA", in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., 2001, pp. 885-888.
- [9] G. Hampson and A. Paplinski, "Phase shift beamforming using cordic", in Proc. Int. Symp. Signal Process. Appl., 1996, pp. 684-687.
- [10] A. Trucco, "A least-squares approximation for the delays used in focused beamforming", J. Acoust. Soc. Amer., vol. 104, no. 1, pp. 171-175, Jul. 1998.
- [11] J. E. Volder, "The CORDIC trigonometric computing technique", IRE Trans. Electron. Comput., vol. EC-8, no. 3, pp. 330-334, Sep. 1959.
- [12] J. S. Walther, "A unified algorithm for elementary functions", in Proc. Spring Joint Comput. Conf., pp. 379-385.
- [13] A. Trucco, "Enlarging the scanning region of a focused beamforming system", Electron. Lett., vol. 33, no. 17, pp. 1502-1504, Aug. 1997.
- [14] B. O. Odelowo, "A fast beamforming algorithm for planar/volumetric arrays", in Proc. 39th Asilomar Conf. Signals, Syst. Comput., 2005, pp. 1707-1710M.
- [15] Palmese and A. Trucco, "Acoustic imaging of underwater embedded objects: Signal simulation for three-dimensional sonar instrumentation", IEEE Trans. Instrum. Meas., vol. 55, no. 4, pp. 1339-1347, Aug. 2006.