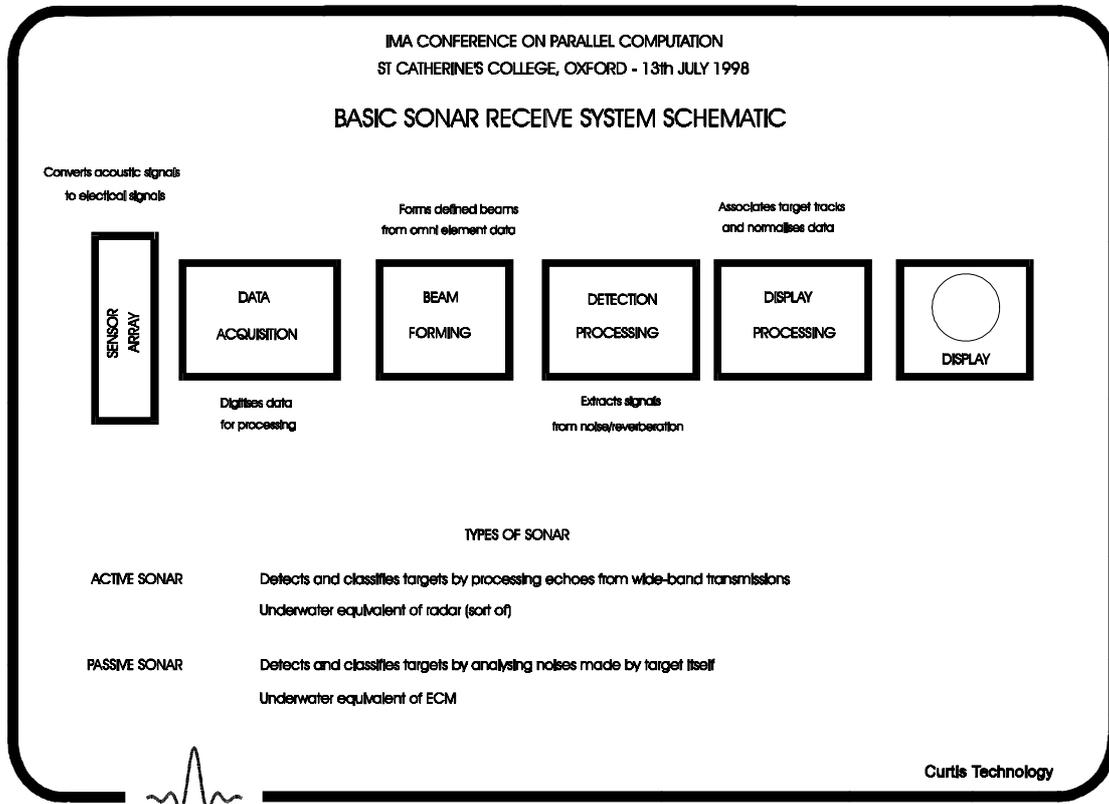


PRINCIPLES OF SONAR BEAMFORMING

This note outlines the techniques routinely used in sonar systems to implement time domain and frequency domain beamforming systems. It takes a very simplistic approach to the problem and should not be considered as definitive in any sense.



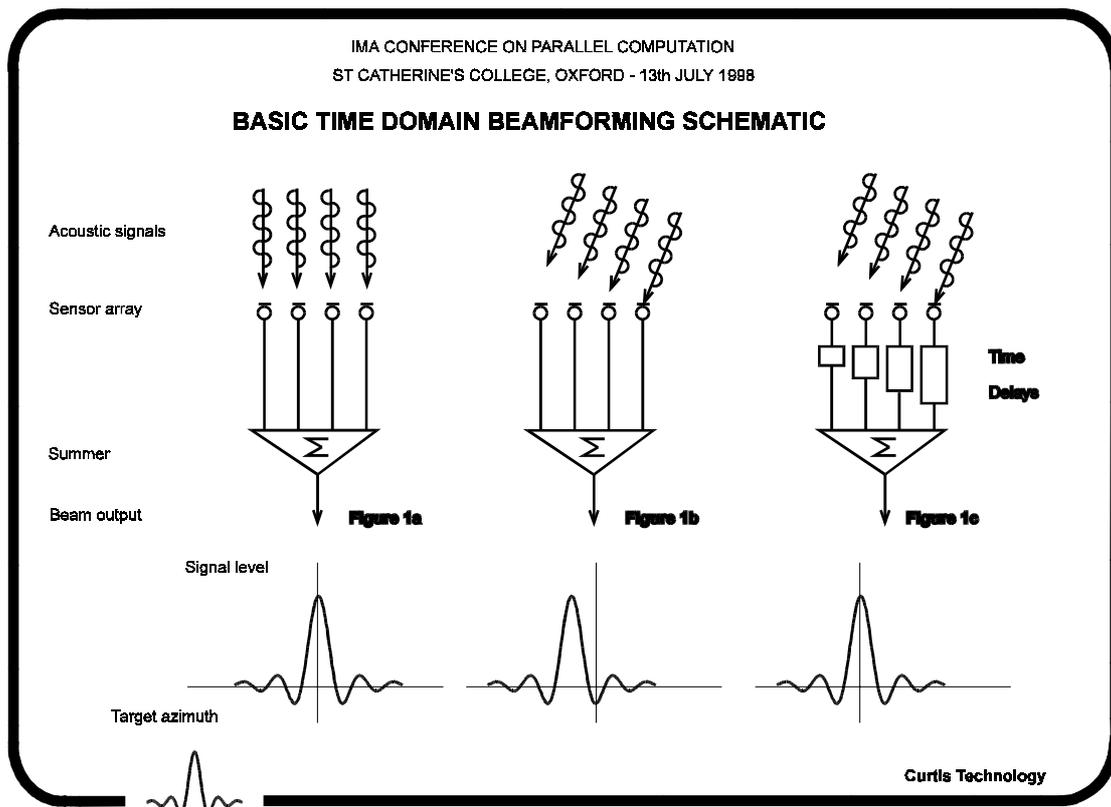
TIME DOMAIN SONAR BEAMFORMING.

Consider an array of hydrophones receiving signals from an acoustic source in the far-field (Figure 1a). If the outputs from the 'phones are simply added together then, when the source is broad-side to the array, the 'phone outputs are in phase and will add up coherently.

As the source is moved around the array (or the array rotated), then 'phones across the array receive signals with differential time delays, so the 'phone outputs no longer add coherently and the summer output drops (Figure 1b).

If we plot the signal level as the array is rotated, we get the array beam-pattern: in this case, where all 'phone outputs are weighted uniformly, a classical sinc function. In order to reduce the effects of the spatial side-lobes of the beam-pattern, an array shading function is applied across the array (often a Hamming or Hanning weighting).

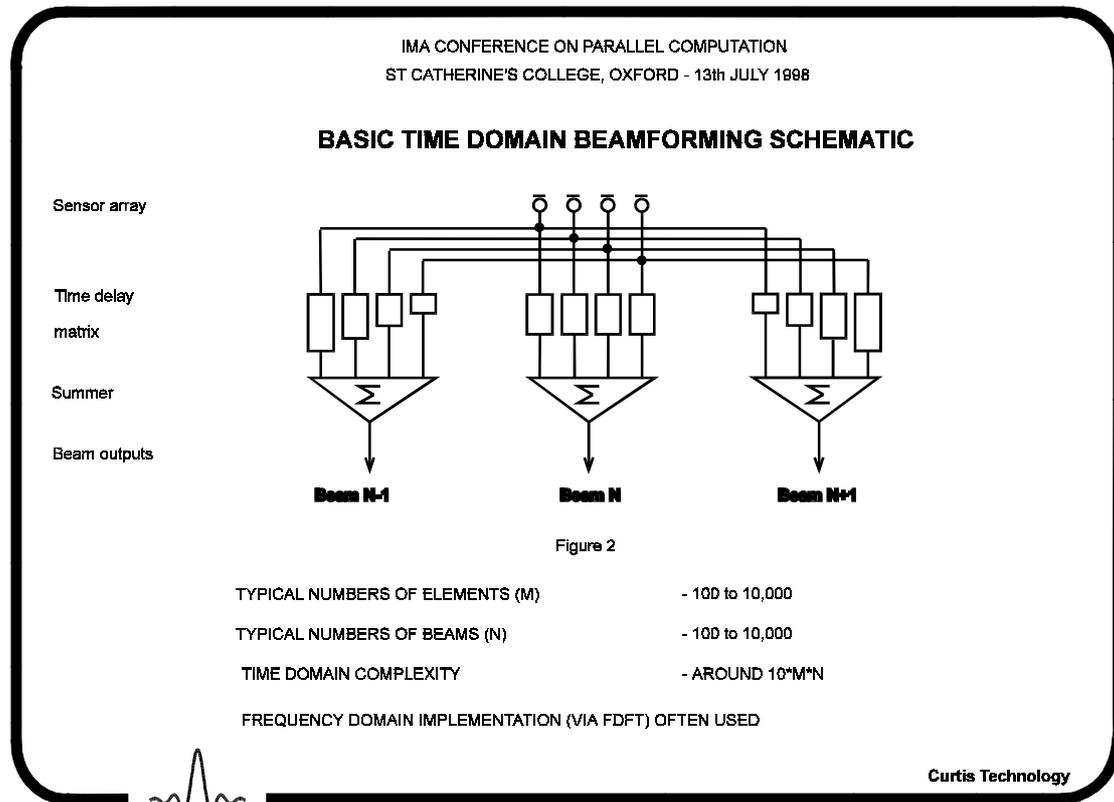
In radar beamforming systems, this weighted element summation is often all that is needed. The antenna can then be rotated to scan the narrow beam formed by adding together the array elements. However, in sonar we can not usually use this method. Firstly, sonar arrays are often pretty big and heavy so must be mounted on the hull of a ship or submarine. Secondly, even if we could rotate them, as the array was turned, the movement of water across the array face would generate flow-noise. This would swamp out the received signals we were trying to detect. Consequently, sonar arrays are usually mounted in a fixed position on a platform (or towed behind it) and scanned electronically.



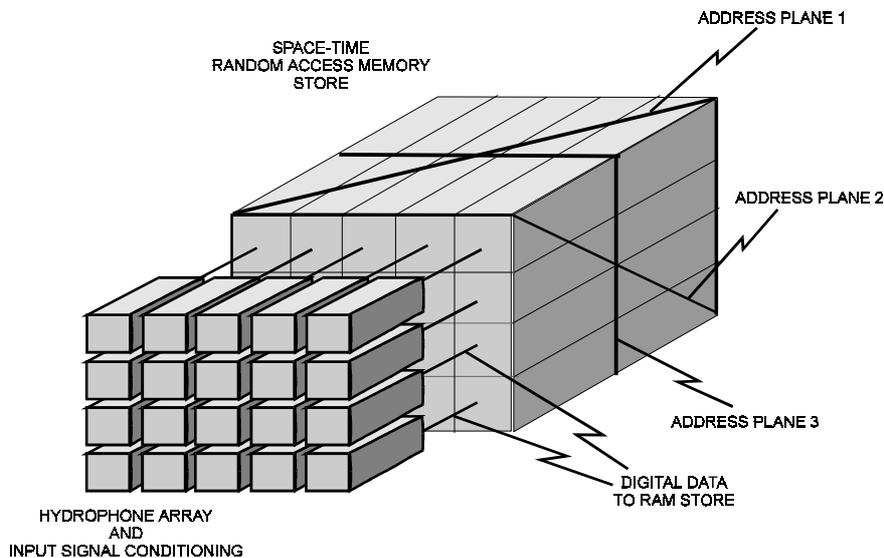
Electronic beam-steering can be achieved by introducing a time delay network between the individual 'phone outputs and the beam summer, so that signals from the required 'look direction' are brought into phase and can be added together coherently (Figure 1c). This is similar to phased array radar.

Conceptually, we could systematically vary these time delays to electronically scan a single beam around the platform, but again we have problems. If we consider an active system (where acoustic energy is transmitted from the platform and targets located by detecting echoes), it takes around 2 minutes for the sound energy to get from the transmitter, out to say 50 kYards, and for any echoes to propagate back (speed of sound in water is around 1500 metres/second). During this time we need to maintain the receive beam looking in the same direction, so as not to miss any potential echoes. So for a typical beamwidth of say around 1 degree, we would need to step the receive

beam around in 1 degree, dwelling for two seconds every time we transmit to receive any echoes. To completely search 360 degrees around the ship would take around 12 minutes!! This obviously is not sensible (a torpedo attack can be all done in around 35 seconds), so a number of delay/summation networks are used in parallel to form a fan of beams with respect to the array (Figure 2).



Although the delay networks are shown separately in Figure 2, in practice they usually use a common random access memory store [1]. This store is organised to hold a running time history of the acoustic data received by the array (Figure 3). There is a one-to-one mapping between the element position in the array and where that data is stored in the memory. The store is updated, usually by sampling data from all elements simultaneously at several times the Nyquist rate. In between write updates, samples of the required output beams are generated sequentially. These are formed by adding together data accessed by addressing planes across the RAM space-time matrix. For example, address plane 1 provides a systematic time delay along the array and forms a beam to endfire. Address plane 2, a delay down the array to form a downward looking beam. Address plane 3, equal delay in all channels to form a beam normal to the array in both azimuth and elevation.



It is an easy step from here to stabilise the beams in space by compensating for the effects of platform motion. If the array motion is monitored, using for example a set of tri-axial accelerometers, this motion data can be used to correct the read address planes to compensate for the movement and to inertially stabilise the beams in space. It is also an easy step to generalise the space-time beamformer to handle other array geometries, for example to handle line, cylindrical or conformal arrays, by simply mapping the planar addresses onto the more complex array geometry (using PROM look-up for example).

In summary time domain beamforming based on space-time RAM stores is very flexible and widely used. The main problem lies in the overall amount of hardware needed to use this type of system. In practice, to maintain good side-lobe levels, the time resolution used to form beams must be equivalent to sampling the 'phone data at around 10 times the Nyquist rate (high rate required to reduce time quantisation effects on spatial side-lobes). This can be achieved either by heavily oversampling the 'phone data or by sampling at a lower rate (maybe down to close to Nyquist) and then interpolating the data, often using FIR interpolators, to improve the time resolution of element data into the beam summer. This interpolation can be carried out either before or after the delay/storage operation: in practice a combination of pre-store and post-store interpolation is used. Either way, the need to oversample or to interpolate increases processing load and for large systems, frequency domain realisations are often used to minimise system size and cost. But as always there are no free dinners, the cost in the case of frequency domain beamformers is in the added complexity of the algorithms.

FREQUENCY DOMAIN SONAR BEAMFORMING.

The main aim of using frequency domain techniques for sonar beamforming is to reduce the amount of hardware needed (and hence minimise cost). The time-domain system outlined above is very flexible and can work with non-equi-spaced array geometries. It is very efficient with arrays with small numbers of channels, say up to 128 channels, but as it is essentially an $O(N^2)$ process it becomes unwieldy with large arrays.

Many sonar systems need to use spectral data: for example, in active pulse-compression systems, the correlation processing is often conveniently carried out using fast frequency domain techniques and for passive systems data is usually displayed as a spectrum versus time (LOFARgram) plot for a number of look directions. In these types of system, it may be convenient to use frequency domain beamforming to avoid some of the time-frequency, frequency-time transformations that would be needed if time domain beamforming were used. There are several classes of frequency domain beamforming:-

1. 'conventional' beamforming, where array element data is essentially time delayed and added to form beams, equivalent to a spatial FIR filter,
 2. adaptive beamforming, where more complicated matrix arithmetic is used to suppress interfering signals and to obtain better estimates of wanted targets
- and
3. high resolution beamforming, where in a very general sense target signal-to-noise is traded to obtain better array spatial resolution.

Here we will consider only conventional beamforming. The conventional beamforming algorithms can be again sub-divided into three classes, narrow-band, band-pass and broad-band systems.

NARROW BAND SYSTEMS.

If the beamformer is required to operate at a single frequency, then the time delay steering system outlined above can be replaced by a phase delay approach. For example, the time domain beamformer output can be written as:-

$$A_r(t) = \sum_{k=0}^{k=N-1} W_k \cdot f_k(t + \tau_{k,r})$$

where N is number of hydrophones in the array

W_k is the array shading function

$f_k(t)$ is the time domain k^{th} element data

and $\tau_{k,r}$ is the time delay applied to the k^{th} element data for the r^{th} beam

If $f(t)$ is generated by a narrow band process, then we can write:-

$$\begin{aligned} f(t) &= \cos(\omega t) \\ &= \operatorname{Re} [\exp\{-j\omega t\}] \end{aligned}$$

If we take a snap-shot of data across the array at some time $t=T_0$ when the source is at some angle Φ with respect to broad-side (assume an equi-spaced line array), then we have:-

$$\begin{aligned} f_k(T_0) &= \cos(\omega T_0 + k\phi) \\ &= \operatorname{Re} [\exp\{-j(\omega T_0 + k\phi)\}] \end{aligned}$$

where ϕ is the differential phase across successive elements in the array due to the relative bearing Φ of the source.

We can bring the element data into phase by correcting for this $k\phi$ term, when the array beam sum output is steered to look towards the source at bearing Φ . If we form our beam sum output as:-

$$\begin{aligned} A_r(T_0) &= \sum_{k=0}^{k=N-1} W_k \cdot f_k(T_0) \cdot \exp\{-jk\phi\} \\ &= N \cdot \cos(\omega T_0) \end{aligned}$$

In practice, we know we want to form a fan of say M beams from the array, so we can write:-

$$A_r(T_0) = \sum_{k=0}^{k=N-1} W_k \cdot f_k(T_0) \cdot \exp\{-jkr\theta\} \quad \dots \quad \text{for } -M/2 \leq r \leq M/2-1$$

If we choose θ so that $\theta \cdot M/2$ is equal to the differential phase between elements when the source is at end-fire, then we have formed a fan of M beams covering ± 90 degrees about broad-side.

This process is repeated on successive snap-shots of array data, with snapshots gathered at some rate faster than Nyquist for the frequency of interest.

If we compare this beam equation with that for the discrete Fourier transform (DFT) of a block of time series data B_k given by:-

$$A_s = \sum_{t=0}^{t=N-1} W_t \cdot B_t \cdot \exp\{-j2\pi/N \cdot st\} \quad \dots \text{ for } -N/2 \leq s \leq N/2-1$$

then it can be seen that the two are similar and we can re-write the beamforming algorithm as:-

$$A_r(T_0) = \sum_{k=0}^{k=N-1} W_k \cdot f_k(T_0) \cdot \exp\{-j2\pi/N \cdot kr\alpha\} \quad \dots \text{ for } -N/2 \leq r \leq N/2-1$$

This is identical to the DFT equation, except for the α term in the complex exponent: α is usually between zero and one for beamforming.

In the DFT case, the coefficients are based on the integral roots of unity, $\exp\{-j2\pi/N\}$, whereas, in the beamforming case, the coefficients use the fractional roots of unity, $\exp\{-j2\pi/N \cdot \alpha\}$. Hence the beamforming equation is a fractional discrete Fourier transform (FDFT) [2] rather than a DFT. The introduction of this factor α removes the symmetry of the basic DFT and at first sight it would appear that an order N^2 process is needed to realise the algorithm. However, there are several techniques that allow the FFT to be used to approximate the required FDFT, particularly for the narrow-band signal case outlined here.

Firstly, one can pad the input data sequence (the snap-shot of data across the array) by appending zeroes to the data block – this is effectively an interpolation process that allows the FFT exponents used for transforming the input data to approach the required FDFT exponents. The second approach is to use the nearest larger convenient FFT block size and to select the transform outputs closest to those that would have been generated by the exact FDFT algorithm. Both of these approaches are used in practice and result in narrow-band beamformers with order $N \cdot \log N$ processes. However, there are fast algorithms for the FDFT, similar to the chirp-Z transform, and in many applications, particularly broad-band systems, these are more useful.

WIDE-BAND FREQUENCY DOMAIN BEAMFORMING [5]

The frequency domain systems outlined above rely on the source being narrow band. This is not usually the case in sonar, although it is often a reasonable approximation in radar and some communication systems.

One obvious way to extend the narrow band implementations is to gather a block of data from each sensor in the array (rather than snapshot across the array) and to use an FFT to convert each block into the frequency domain to generate narrow band components. The narrow band beamforming algorithms outlined above can then be applied sequentially to each of the frequency components in turn. This approach, a 2D FFT beamformer, is shown schematically in Figure 4 below. The system complexity of this type of system is much less than the corresponding time domain implementation for large arrays but there are some problems.

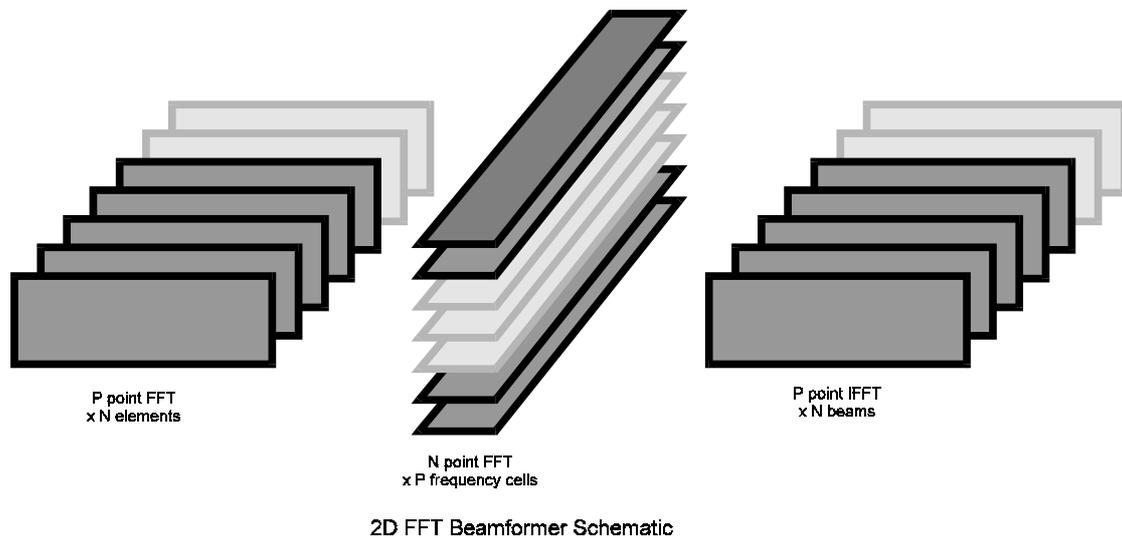


Figure 4 – 2D FFT Beamformer Schematic

Blocks of time domain element data are first transformed into the F-domain using P point FFTs. N point FFTs are performed across the array for each frequency cell of the P point FFTs to form N 'beams' each with P frequency cells. The P frequency cells per 'beam' are then transformed back to time domain using P point IFFTs.

If either of the simplistic approximations to the FDFT are used for the beamforming part of the process then, because the required value of α changes linearly with frequency, the effective maximum response axis (MRA) of any beam from the system is also frequency dependent. This effect is shown in Figure 5.

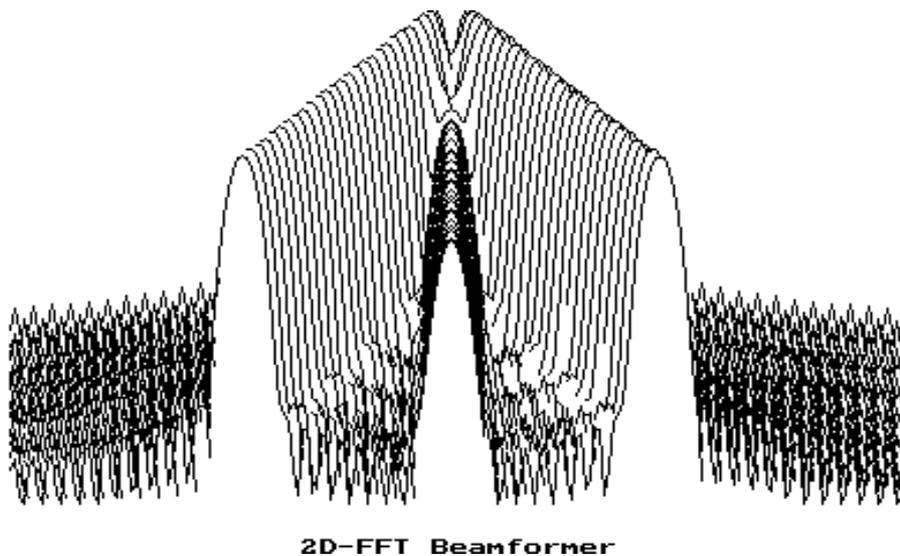


FIGURE 5 – Output from a 2D FFT wide-band beamformer.

X axis corresponds to bearing, Y axis signal level and the Z axis frequency. High frequencies are at the front of the plot with low frequencies at the back. Note that received bearings change with frequency.

This shows the bearing/frequency distribution for a 2D FFT beamformer receiving three broad-band contacts at +45, 0 and –45 degrees wrt array broad-side. It can be seen that the beam MRAs for targets off broad-side vary linearly with frequency. This creates considerable problems in down-stream processing: systems usually require that beam pointing directions are independent of frequency. (The reason for this frequency/bearing variation is due to the fact that the 2D FFT beamformer actually transforms element data into wave-number/frequency space rather than into bearing/frequency space).

A number of tricks have been used in the past to correct for this MRA variation. If the received data is relatively narrow-band, it is often ignored. With broader-band signals, an interpolation process can be used to interpolate the 2D FFT output into beams. The problem with this approach is that the interpolation process usually requires 2D FIRs and the interpolation scheme is often more complex than the time domain beamforming process that it is trying to replace!

The only real way to make the process broad-band is to tackle the problem of finding a fast algorithm for the FDFT. Then the beamforming process can be made exact, with the value of α changed exactly for each frequency component in the broad-band signal, thus producing beams with frequency invariant MRAs.

FAST ALGORITHMS FOR THE FRACTIONAL DFT.

We have shown above that the narrow band beamformer can be implemented by using an algorithm of the form:-

$$A_{r,\omega} = \sum_{k=0}^{k=N-1} W_k \cdot f_k \cdot \exp\{-j2\pi/N \cdot kr\alpha\} \dots \text{for } -N/2 \leq r \leq N/2-1$$

where α is a function of ω , the frequency cell currently being processed.

We will combine the weighting function with the data sample and simplify the equation to write:-

$$A_{r,\omega} = \sum_{k=0}^{k=N-1} B_k \cdot \exp\{-j2\pi kr\beta\} \dots \text{for } 0 \leq r \leq N-1$$

where β is equal to α/N .

Using Bluesteins decomposition [3], we can write $2kr = k^2 + r^2 - (r - k)^2$, giving:-

$$\begin{aligned} A_{r,\omega} &= \sum_{k=0}^{k=N-1} B_k \cdot \exp\{-j\pi [k^2 + r^2 - (r - k)^2]\beta\} \\ &= \exp\{-j\pi r^2 \beta\} \sum_{k=0}^{k=N-1} B_k \cdot \exp\{-j\pi k^2 \beta\} \cdot \exp\{-j\pi (r - k)^2 \beta\} \\ &= \exp\{-j\pi r^2 \beta\} \sum_{k=0}^{k=N-1} Y_k \cdot Z_{r - k} \end{aligned}$$

where $Y_k = B_k \cdot \exp\{-j\pi k^2 \beta\}$

and $Z_k = \exp\{j\pi k^2 \beta\}$

The summation term is the discrete convolution of Y_k and Z_k and can be calculated using one of the usual fast frequency domain approaches [4]. This has the minor complication that the FFT methods generate a circular convolution, so we need to extend the length of the two sequences, by padding the data blocks with zeroes to some length q , where q is the nearest convenient FFT block size greater than or equal to $2N$. Then we can implement the FDFT using the block schematic shown in Figure 6, with the complete broad-band beamformer schematic is as shown in Figure 7.

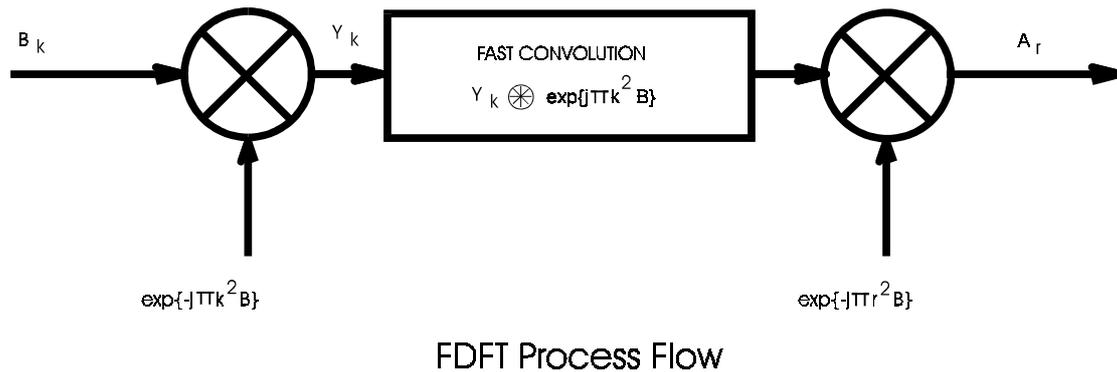


Figure 6 – Practical Implementation of fast Fractional DFT.

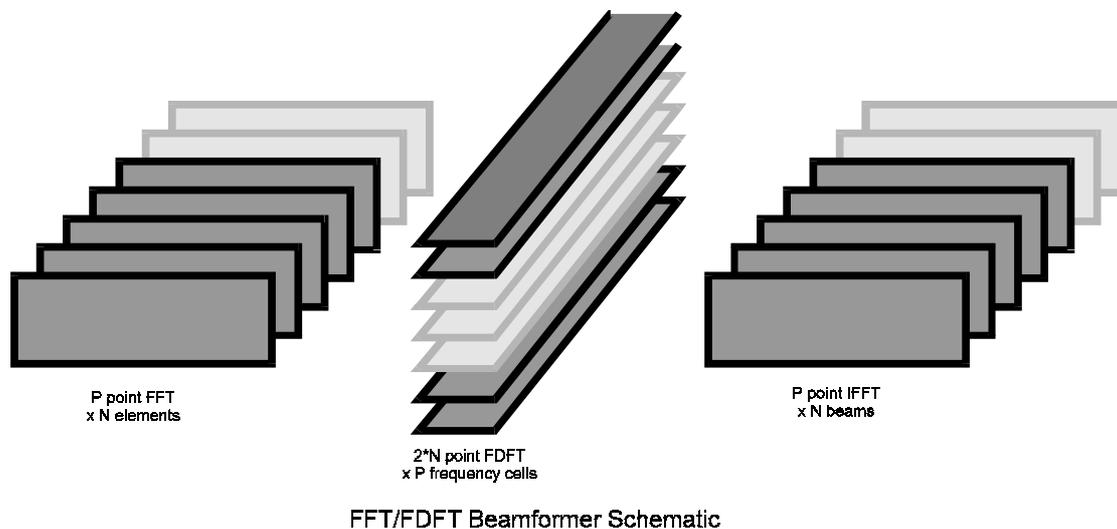


Figure 7 – FFT/FDFT Beamformer Schematic

Beam plots from this process for the same target scenario as shown previously are in Figure 8 – it can be seen that beam MRAs are invariant with frequency and that beam widths increase as frequency decreases. This type of frequency domain system generates beam outputs identical to a time domain beamforming system, i.e. it is a frequency domain implementation of a time domain process, rather than an approximation. The technique has been widely used in UK sonar systems and has been generalised for planar, cylindrical, conical and spherical array geometries.

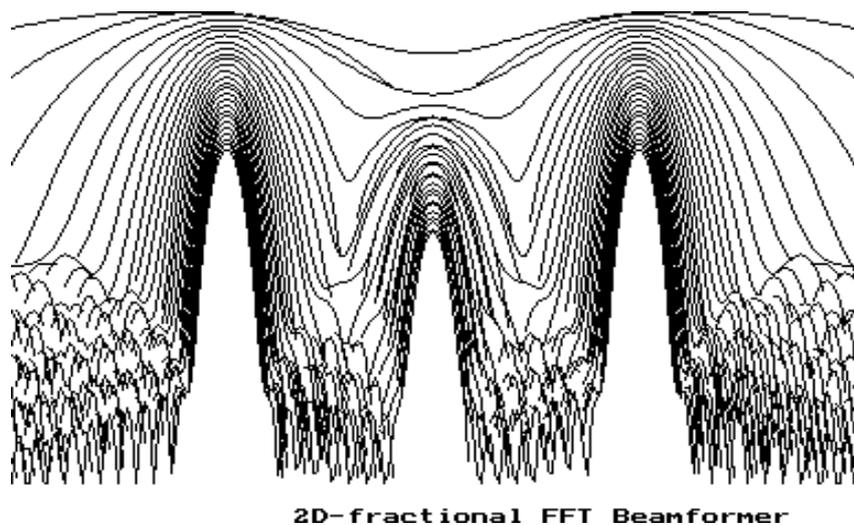


Figure 8 – Output from an FFT/FDFT Beamformer.

*X axis corresponds to bearing, Y axis to signal level and Z axis frequency.
Note that beam MRAs are invariant with frequency.*

The overall efficiency of the process is dependent on an efficient FFT implementation: higher radix transforms in general provide better performance. In general, the broad-band frequency domain approach requires considerably less hardware than the equivalent time domain system when processing arrays of more than around 100 elements. Below 32 elements, the time-domain method probably wins, but it may well still be more convenient to use frequency domain beamforming if data must be transformed into the F-domain for other operations, e.g. in LOFAR systems, or when fast F domain replica correlation is used.

Copyright Curtis Technology (UK) Ltd 1998.

REFERENCES

1. T E Curtis and R J Ward, 'Digital Beamforming for Sonar', IEE Proc., Part F, Comms., Radar and Signal Processing, Vol 127, 1980.
2. D H Bailey and P N Swartztrauber, 'The Fractional Fourier Transform and Applications', SIAM Review, Vol 33, No 3, pp 389-404, Sept 1991.
3. L I Bluestein, 'A Linear Filtering Approach to the Computation of the Discrete Fourier Transform', IEEE Trans. Audio Electroacoust., Vol 18, pp 451-455, 1970.
4. R C Agarwal and J W Cooley, 'New Algorithms for Digital Convolution', IEEE Trans Acoust. Speech. Signal Process., Vol 25, pp 392-410, 1977.
5. Tom Curtis, et al, 'Wide-band, High Resolution Sonar Techniques', IEE Colloquium on Underwater Applications of Image Processing, London, 25 March, 1998 - IEE Ref No 1998/217