Automatic dynamic focusing through interfaces

J. F. Cruza, J. Camacho, J. M. Moreno, C. Fritsch

Ultrasound for Medical and Industrial Applications Group (UMEDIA)
Spanish National Research Council (CSIC)
La Poveda (Arganda del Rey), 28500 Madrid, Spain

Abstract – **An interface between the coupling medium and the inspected part is frequently found in Non Destructive Testing (NDT). When phased array technology is used, the focal laws must be obtained considering the refraction at the interface, which requires computationally intensive iterative procedures since no closed formulae exist for the general case.**

This work presents a new two-step approach. In the first step, the two propagation media scenario is converted into a single homogeneous medium by computing a *virtual array* with nearly equivalent flight times to the foci. In the second step, a *focusing hardware*, conveniently initialized, evaluates in real-time the sampling instants that correspond to the focal laws.

Focusing errors are small enough to validate the new technique for a wide range of applications. In fact, for active apertures currently used in NDT, the resolution and dynamic range are almost not affected as it is experimentally shown. Furthermore, focal law computing time is dramatically reduced by evaluating a few parameters instead of focusing delays for every output sample, element and steering direction. The instrument focal law storage requirements become significantly reduced as well.

Keywords **–** *Phased array*, *dynamic focusing, interfaces, NDT.*

I. INTRODUCTION

The phased array technique finds many applications in Non Destructive Testing (NDT) due to its capabilities of electronic beam steering and focusing. Dynamic focusing is being progressively included in the NDT phased array instruments due to its superior image quality.

However, dynamic focusing requires a large number of focal laws, ideally one for every output sample in the beamformed image. This demands a large amount of memory in the phased array instrument, although it can be reduced by coding [1-2]. Furthermore, in many NDT applications, there is a coupling medium (a wedge, water, etc.) between the array transducer and the inspected part, whose shape can be arbitrary.

Obtaining the focal laws for the most general case (coupling medium, geometry, etc.) is a computationally intensive process that must take into account refraction at the interface. There are not closed formulae and simulation tools or numerical iterative procedures must be used [3-5]. This usually involves a large computing time to get the focal laws for all the samples and steering angles.

This work presents a new two-step approach. In the first one, the two propagation media scenario (coupling and inspected part) is converted into a single homogeneous medium. To this purpose, a *virtual array* that operates in the second medium only is obtained. Flight times to the foci are nearly equivalent. In the second step, a *focusing hardware*, conveniently initialized, evaluates in real-time the dynamic focusing sampling instants from the virtual array. Although any of the known circuits that operate in homogeneous media for dynamic focusing could be used (for instance, [6-9]), the proposed hardware has some features that are interesting for its implementation in state-of-the-art FPGAs.

This methodology provides several advantages: a) It avoids the cumbersome process of computing focal laws through arbitrary geometry interfaces taking into account the refraction laws; b) Memory requirements in the phased array instrument are significantly reduced; c) Strict dynamic focusing (at all the output samples) is achieved in real-time; d) Parameter computing and uploading to hardware is very fast; e) the focusing hardware is simple enough to be efficiently integrated within the beamformer.

II. THE VIRTUAL ARRAY

The idea is to obtain a virtual array that, operating in the second medium only, avoids the problems associated with the refraction at the interface and provides approximately equal flight times to the foci. Fig. 1 shows the general arrangement, with an arbitrarily shaped interface between two media with sound velocities c_1 and c_2, respectively. In this figure, the resulting virtual array is also shown. Foci are located in *main rays* forming an angle θ with the normal to the interface. The incident ray has an angle β that results from the Snell's law. From these data, the steering angle α and the emission focal law can be computed.

The time-of-flight from focus F to array element A is:

$$t_F = r_{AG} / c_1 + r_{GF} / c_2 \qquad (1)$$

where the difficulty is on computing the coordinates of the entry-point G. But, if the curvature radius of the interface is large enough, the application of the Abbe's invariant of geometric optics yields:

$$r_{VG} \approx r_{AG} c_1 / c_2 \qquad (2)$$

With $t_{AG} = r_{AG}/c_1$ and $t_{VG} = r_{VG}/c_2$,

$$t_{VG} = (c_1^2 / c_2^2) t_{AG} \qquad (3)$$

Considering that the virtual array operates in the second medium with propagation velocity c_2, $t_{VF} \approx t_{VG} + t_{GF}$. Then,

$$t_{AF} \approx t_{VF} + t_{AG}(1 - c_1^2 / c_2^2) \qquad (4)$$

Thus, the time-of-flight from array element A to focus F is approximately equal to that from the corresponding virtual array element V to F plus a constant term t_K:

$$t_K = t_{AG}(1 - c_1^2 / c_2^2) \qquad (5)$$

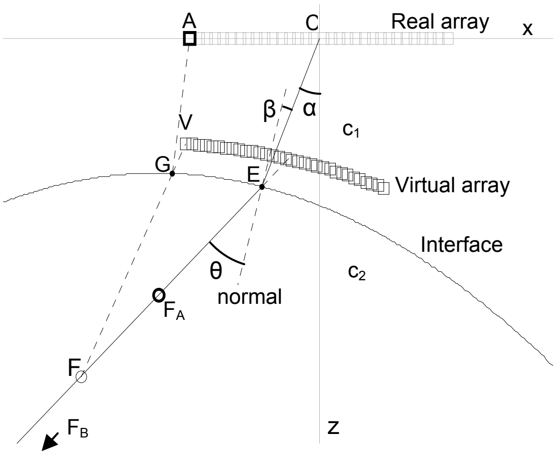

Fig. 1. Geometry. Representation of the real and virtual arrays, the latter operating in the second medium only of propagation velocity c$_2$.

The virtual array coordinates are found by solving (4) for two foci, F$_A$ and F$_B$, conveniently located in the main ray near and far from the interface, respectively. The flight times to these foci are computed searching for the minimum of (1) in application of the Fermat's principle.

III. Focusing Hardware

Setting the coordinates origin at the virtual array centre (Fig. 2), the distance r_F from virtual element at (x_V, z_V) to a focus located at (R_F, θ) is:

$$r_F = \sqrt{R_F^2 + x_V^2 + z_V^2 - 2R_F(x_V \sin\varphi + z_V \cos\varphi)} \qquad (6)$$

The sampling period T_S defines the interval ΔR between samples, $\Delta R = c_2 T_S/2$. Normalizing all the distances by ΔR, we get $n = R_F/\Delta R$=integer, $x = x_V/\Delta R$, $z = z_V/\Delta R$ and $r_n = r_F/\Delta R$:

$$r_n = \sqrt{n^2 + x^2 + z^2 - 2n(x \sin\varphi + z \cos\varphi)} \qquad (7)$$

Squaring this equation with $\alpha = x\sin\varphi + z\cos\varphi$, $\beta = x^2 + z^2$:

$$r_n^2 = n^2 - 2n\alpha + \beta \qquad (8)$$

$$r_{n+1}^2 = r_n^2 + 2n + H \qquad (9)$$

where H=1-2α is constant for an element and steering angle.

It has been shown [2] that, for an aperture size D_V and from a minimum range R_0, the differences in the round-trip time-of-fligth to consecutive foci, $\Delta t_n = t_{n+1} - t_n$ verify:

$$1 - 1/v \le t_{n+1} - t_n < 1, \quad n \ge n_0 \qquad (10)$$

where v is a constant greater than unity, $n_0 = R_0/\Delta R$ and,

$$R_0 = vD_V / (4\sqrt{v-1}) . \qquad (11)$$

For example, choosing v=10 results in $0.9 \le \Delta t_n < 1$ from a relative range R_0/D_V =0.83. The hardware must find an estimate $\hat{t}_n$ of t_n such as $|t_n - \hat{t}_n| < 1/v$, that is, a fraction of a sampling period, where:

$$t_n = (n + r_n)/2 \qquad (12)$$

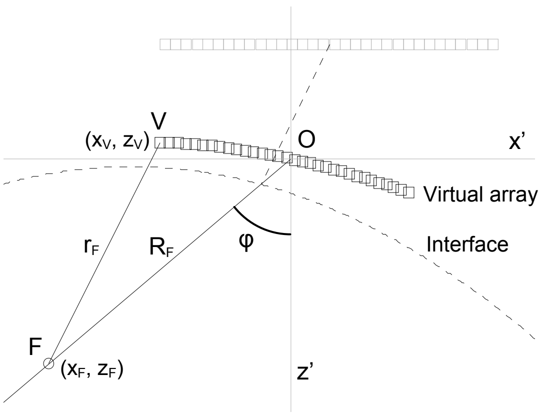

Fig. 2.- Traslation of the coordinates origin to the virtual array centre.

Substitution in (10) yields,

$$r_n + 1 - \frac{2}{v} \le r_{n+1} < r_n + 1 \qquad (13)$$

A binary variable Q_n is evaluated for every sample, $Q_n = (r_n < \hat{r}_n)$. The next estimation of r_n to keep the errors within the bounds of (10) and (13) becomes:

$$\hat{r}_{n+1} = \hat{r}_n + 1 - 2Q_n / v = \hat{r}_n + P_n \qquad (14)$$

$$\hat{r}_{n+1}^2 = \hat{r}_n^2 + 2P_n\hat{r}_n + P_n^2 \qquad (15)$$

Subtracting this equation from (9),

$$r_{n+1}^2 - \hat{r}_{n+1}^2 = (r_n^2 - \hat{r}_n^2) + 2n - 2P_n\hat{r}_n + H - P_n^2 \qquad (16)$$

Now, let us call:

$$\begin{aligned} B_n &= 2n + H \\ C_n &= P_n(2\hat{r}_n + P_n) \\ D_n &= r_n^2 - \hat{r}_n^2 \end{aligned} \qquad (17)$$

that can be iteratively computed as:

$$\begin{aligned} B_{n+1} &= B_n + 2 \\ D_{n+1} &= D_n + B_n - C_n \\ \hat{r}_{n+1} &= \hat{r}_n + P_n \end{aligned} \qquad (18)$$

Since both r_n and $\hat{r}_n$ are positive magnitudes, the binary variable Q_n can be also computed from:

$$Q_n = (r_n^2 < \hat{r}_n^2) = (D_n < 0) \qquad (19)$$

The binary variable Q_n tracks the focusing time at every sample by advancing by 1-1/v the sampling instant if Q_n=1 or leaving a sampling period to the next sample if Q_n=0, keeping the timing error within $\pm T_S/v$. Q_n is just the sign of D_n, whose value can be iteratively obtained from (18).

Initial values for the variables involved are set for $n = n_0$, obtained from (11) and from the definition of α, β and H. The initial value D_0 can be =0, so that only 2 parameters have to be preset for every element and steering angle.

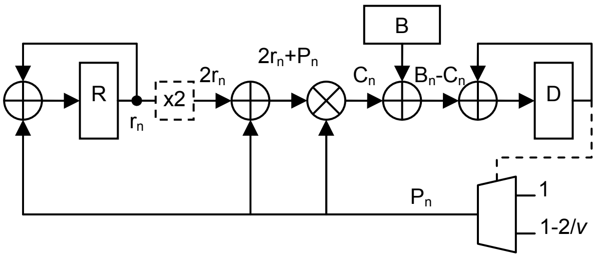

Fig. 3. Focusing hardware. Register R holds $\hat{r}_n$, counter B holds B_n and register D holds D_n. The multiplexer is controlled by the sign of D.

The circuit shown in Fig. 3 implements the algorithm described above. It has 3 registers for $\hat{r}_n$, B_n and D_n, 4 adders and a multiplier. A multiplexer provides P_n=1 or P_n=1-1/2v as required by the sign of D_n, which is the focusing code Q_n.

Not shown is a bank of fractional delay filters that produces samples at intervals 1/v the sampling period. This filter bank is controlled by Q_n, as performed in [10] to keep track of the dynamic focusing.

Note that the operation "x2" requires no logic. Although the multiplier could be easily substituted by a multiplexer and an adder if v is power of 2, it provides two advantages: first, no restrictions apply to the chosen value of v and, second, state-of-the art FPGAs include DSP cells. The architecture shown adapts well to these cells, thus avoiding the use of distributed logic, which results in lower power consumption and the possibility of sharing a single focusing circuit among several channels.

IV. Performance

The main source of timing errors is the approximation of t_K given by (5) being constant. In fact, the entry-point G varies with the focus position, so that t_{AG} is not constant. Since the virtual element V is in the prolongation of the segment FG, its position should be modified for every focus. However, in most configurations, these variations are small enough to produce no significant errors.

Another source of error is that (4) is solved for two foci, F_A and F_B, where timing errors are zero. In their proximities it is expected that these errors are low. Setting F_A and F_B near the origin and end of the acquisition range, respectively, is a good trade off.

On the other hand, the focusing hardware has a timing error within $\pm T_S/v$ following from the algorithm derivation.

It is not easy to obtain the bounds of the timing error for the general case of arbitrarily shaped interfaces, steering angles, distance from the array, sound velocity, etc. There are too many variables involved and simulation seems the best tool to evaluate the proposed technique.

To this purpose, the reference time of flight is computed from every sample to real array elements following the Fermat's principle (fastest path), with a finely discretized interface. Then, flight times to the corresponding virtual array elements operating in the second medium only are computed. The differences are considered timing errors.

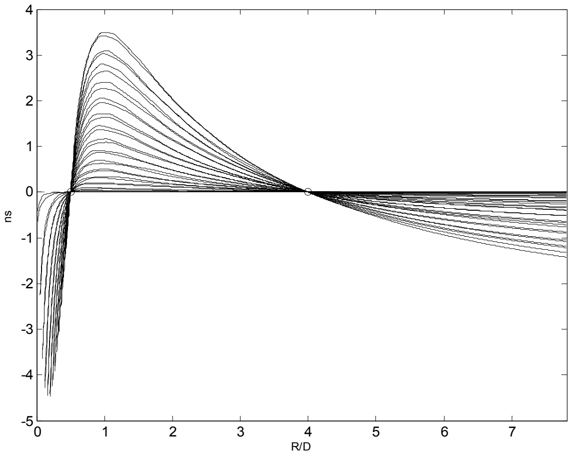

Fig. 4. Timing errors in ns with F_A at 0.5D and F_B at 4D from the interface and c_1 = 1.5 mm/µs, c_2 = 6.2 mm/µs

Fig. 4 shows the timing errors in ns for the configuration shown in Fig. 1, with c_1 = 1.5 mm/µs and c_2 = 6.2 mm/µs. The foci F_A and F_B are located at ranges 0.5D and 4D from the interface, respectively. Timing errors are null at F_A and F_B, as it was expected. In the lower ranges, errors are computed for the active elements only, by application of the active aperture given by (11).

It is seen that the maximum error up to, at least, a range of 8D is bounded within ±4 ns. This is a good figure for common situations. For a 5 MHz array, the maximum error is within 1/50 the fundamental period. In this case, with a sampling frequency of 40 MHz and v=4, the focusing circuit error will be bounded within ±6.25 ns, a similar figure. Both errors are independent and the maximum overall error will be about 10 ns, 1/20 the fundamental period.

The effect of the position of F_A is shown in Fig. 5. In this case, F_A is located at range D instead of 0.5D, while the remaining parameters are kept. Errors in the region closer to the interface increase slightly (up to nearly 8 ns) and are lower from F_A and ahead.

This has been repeated for many other configurations, including different shaped interfaces, distances, propagation velocities, steering angles, etc. In all the cases, the maximum errors were in the range of a few ns, being the example shown representative of the simulations carried out.

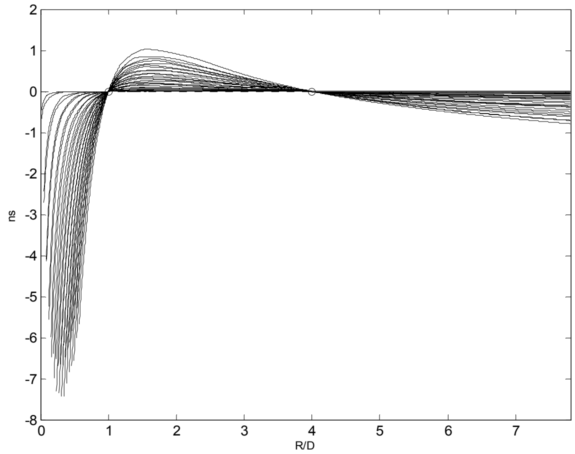

Fig. 5. Timing errors in ns with F_A at 0.5D and F_B at 4D from the interface and c_1 = 1.5 mm/µs, c_2 = 6.2 mm/µs

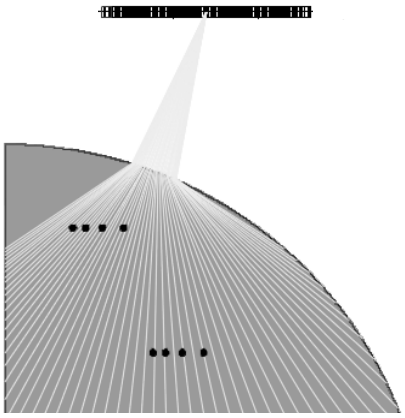

Fig. 6. Schematic for the inspection by immersion of an aluminum cylindrical part with several side-drilled holes

The focusing hardware was implemented in VHDL and integrated on a phased array instrument (SITAU-111, Dasel Sistemas, Spain). A 5MHz, 128-element array with 0.6 mm pitch (Imasonic, Besançon, France) was used to inspect by immersion an aluminum cylindrical part with a curvature radius of 100 mm. A sector scan from -20º to 60º was performed with an active aperture of 64 elements in an arrangement shown in Figure 6.

Figure 7 shows the obtained image, where 8 holes are clearly seen with their correct positions and resolution.

V. Conclusions

A new technique that provides automatic dynamic focusing in presence of interfaces has been proposed. This technique reduces the two media propagation problem to a single one with homogeneous medium in which a virtual array provides nearly equivalent flight times to the foci.

Once the problem is converted to the homogeneous case, any of the real-time focusing circuits previously described in the literature could be applied to perform a strict dynamic focusing at all the acquired samples.

However this work presents a new circuit that has some advantages. Among them, the timing resolution can be arbitrarily set and the circuit can be implemented using the DSP cells available in state-of-the-art FPGAs. The latter provides reduced power consumption. Also, a single focusing circuit can be shared among several channels by time-multiplexing, further reducing the required hardware resources.

Timing errors have been evaluated by comparing the reference time-of-flight from the foci to real array elements with those obtained with the virtual array. This process has been carried out by simulation for many different configurations (steering angles, propagation velocities, interface shape, etc.). In all cases, the errors found are within a few ns or tens of ns, which are good enough to produce high quality images and validate the approach.

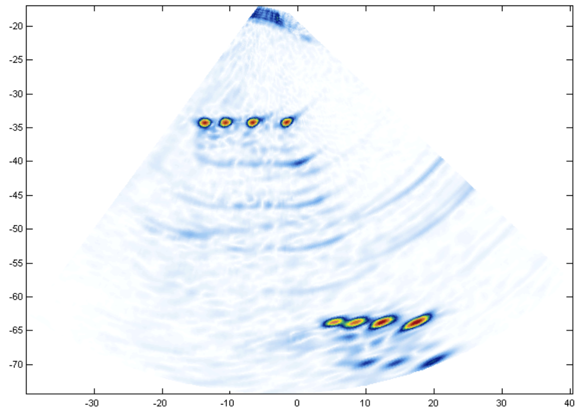

Fig. 7. Image of the SDH in the aluminum cylindrical part obtained with the virtual array concept and the proposed focusing circuit.

Acknowledgements

This work has been supported by Project DPI2010-17648 of the Spanish Ministry of Science and Innovation.

References

[1] D. K. Peterson, G. S. Kino, "Real-Time Digital Image Reconstruction: A Description of Imaging Hardware and an Analysis of Quantization Errors", *IEEE Trans. Sonics Ultrason.*, 31, 4, pp. 337-351, 1984.

[2] C. Fritsch, M. Parrilla, A. Ibáñez, R. Giacchetta, O. Martínez, "The Progressive Focusing Correction Technique for Ultrasound Beamforming", *IEEE Trans. UFFC*, 53, 10, pp. 1820-1831, 2006.

[3] L. Le ber, O. Roy, N. Jazayeri, "Applications of Phased Array Techniques to NDT of Industrial Structures", *2nd. Int'l Conf. on Technical Inspection and NDT* (TINDT'2008), Tehran, 2008.

[4] D. Richard, D. Reilly, J. Berlanger, G. Maes, "Software Tools for the Design of Phased Array UT Inspection Techniques", *SimNDT2010*, http://www.ndt.net/article/SimNDT2010/papers/9_Richard.pdf

[5] C. Poidevin, O. Roy, S. Chatillon, G. Cattiaux, "Simulation tools for ultrasonic testing", *Proc. of the 3rd Int. Conf on NDE in relation to Structural Integrity for Nuclear and Pressurized Component*, Sevilla, 2001.

[6] K. Jeon, M. H. Bae, S. B. Park, S. D. Kim, "An efficient real time focusing delay calculation in ultrasonic imaging systems", *Ultrasonic Imaging*, 16, pp. 231-248, 1994.

[7] M. Bae, "Focusing delay calculation method for real-time digital focusing and apparatus adopting the same", US Pat. 5836881, 17 Nov. 1998.

[8] H. T. Feldkämper, R. Schwann, V. Gierenz, T. G. Noll, "Low power delay calculation for digital beamforming in handheld ultrasound systems", *Proc. IEEE Ultrason. Symp.,* 2, pp. 1763-1766, 2000.

[9] B.G. Tomov. J.A. Jensen, "Compact implementation of dynamic receive apodization in ultrasound scanners", *Proc. SPIE*, 5373, pp. 260-271, 2004.

[10] J. Camacho, M. Parrilla, A. Ibáñez, C. Fritsch, "The Progressive Dynamic Focusing Correction Technique in NDE", *Proc. IEEE Ultrason. Symp*., pp. 1586-1589, 2007.