5 – Array Imaging

Objectives

The purpose of this exercise is to convert raw data measured from an array into an image.

Summary

Modern ultrasonic array controllers can collect the full matrix of raw time-domain signals from an ultrasonic array in a fraction of a second. This is called Full Matrix Capture (FMC). The FMC data contains the time-domain signals from every possible pair of transmitter and receiver elements in an array, so if the array has n elements, it will contain n^2 time-domain signals. Here you are provided with a function that simulates the operation of a real array when it is used on a sample containing three point targets at $(x, y) = (-10, 20)$ mm; $(0, 30)$ mm; $(10, 40)$ mm. The function enables you to specify the parameters of the array (number of elements, element pitch, element width and centre frequency) within certain limits. The task is to write an imaging algorithm that can convert the raw data into an image of the interior of the sample that shows the location of the responses of the targets at the correct locations.

Exercise 5

Download the starter script `STARTER_imaging` and the simulation function `fn_simulate_data_ex5.p` from Blackboard if using Matlab. If using Python, the equivalent function is included in the `NDE_functions.py` module.

If you run the starter script then it will call the simulation function to generate the raw FMC data for an array with the parameters specified in the script. It will then plot one of the raw time-domain signals and a slice through the FMC data so you can see what the raw data looks like. You need to code an imaging algorithm to convert the raw FMC data into a 2D image of the sample that is being inspected. You may also want to pre-filter the raw FMC data prior to imaging to reduce noise. Once you have a working imaging algorithm you can examine the effect of altering the array parameters. You may want to compare the imaging results with the Huygens's field predictions from Exercise 4B as the focal points in that exercise were chosen to coincide with the target points in the sample in this exercise.

You can implement any imaging algorithm you want, but one of the easiest (since you have FMC data) is the Total Focusing Method (TFM):

$$I(\mathbf{r}) = \left| \sum_{T,R} g_{TR}\big(t = \tau_{TR}(\mathbf{r})\big) \right|$$

where $I(\mathbf{r})$ is the intensity of the image at a point $\mathbf{r} = [x \quad y]^T$, $g_{TR}(t)$ is a time-domain signal (see note below) from the FMC data where the subscripts T and R denote the transmitter and receiver element numbers and $\tau_{TR}(\mathbf{r})$ is the delay given by:

$$\tau_{TR}(\mathbf{r}) = \frac{|\mathbf{r}_T - \mathbf{r}| + |\mathbf{r}_R - \mathbf{r}|}{c}$$

where $\mathbf{r}_T$ and $\mathbf{r}_R$ are the positions of the transmitter and receiver element, and c is the speed of sound.

Note: You will get better images if you work with the complex analytic versions of the raw time-domain signal that you can easily obtain by performing a Hilbert transform at the same time as filtering the raw data.

How to use simulation function `fn_simulate_data_ex5`

The data is simulated by the function

`fn_simulate_data_ex5(no_elements, element_pitch, element_width, centre_freq)`

which returns

`time`, `time_data`, and `element_position`.

The inputs are as follows:

- `no_elements` – number of elements in array, must be in range 1 to 64;
- `element_pitch` – pitch of array elements (units: metres);
- `element_width` – width of array elements (units: metres), must be greater than 0.1x10^{-3} m and less that both 4.0x10^{-3} m and (`element_pitch` – 0.05x10^{-3} m) to ensure a minimum gap between elements of 0.05 mm;

- `centre_freq` – centre frequency of array (units: Hertz), must be greater than 1×10^6 and less than 10×10^6.

The outputs are:

- `time` – m-element vector of time (units: seconds),
- `time_data` – an $n \times n \times m$ array (Python) or $m \times n \times n$ matrix (Matlab) of time-traces for every transmitter–receiver combination in array where $n =$ `no_elements`,
- `element_position` – n-element vector of array element positions (units: metres).

The –40 dB bandwidth of the array in the simulation is 112% (i.e. the spectral amplitude is within 40 dB of its peak value at the array centre frequency, f_c, between $0.44f_c$ and $1.56f_c$).