School of Electronic, Electrical and Systems Engineering

Microwave Integrated Systems Laboratory

PhD Thesis

MIMO Sensor Array for Short-Range High-Resolution Automotive Sensing

PhD Student

Mr. Alp Sayin

Supervisors

Prof. Mikhail Cherniakov

Dr. Michael Antoniou

TABLE OF CONTENTS

TABLE OF FIGURES

ACKNOWLEDGEMENTS

I would like to thank my academic supervisors Mike Antoniou and Mike Cherniakov and my industrial supervisor Edward Hoare for their continued support and guidance through my PhD. I am also grateful to them for giving me this opportunity to begin with.

I would also like to thank the rest of the MISL (Microwave Integrated Systems Laboratory) team, for their academic, technical and social support during my research.

In addition, I would also like to thank Jaguar Land Rover and University of Birmingham School of Electrical, Electronics and Systems Engineering for funding my PhD and covering my maintenance.

Finally, I would like to acknowledge the unending love and support of my family. I am indescribably grateful to and for my mother Meral Sayin, my father Erol Sayin and my brother Can Sayin.

LIST OF ABBREVIATIONS

ADC – Analog to Digital Converter

ADS – Almost Difference Sets

AF – Array Factor

API – Application Programming Interface

AWG – Arbitrary Waveform Generator

CDMA - Code Division Multiple Access

CLEAN – Name of a deconvolution procedure to improve resolution proposed by Jan Hogbom

CW – Continuous Wave

DAC - Digital to Analog Converter

DAQ - Nickname of the Data Acquisition Board

DC – Direct Current

DPO – Digital Phosphorus Oscilloscope

FDMA – Frequency Division Multiple Access

FFT – Fast Fourier Transform

FIR – Finite Impulse Response

FMCW – Frequency Modulated Continuous Wave

GMTI – Ground Moving Target Indicator

GA – Genetic Algorithm

GS – Gigasamples

LIST OF ABBREVIATIONS

HP – Hewlett Packard (company)

HPBW – Half Power Beamwidth

IF – Intermediate Frequency

ISLR – Integrated Sidelobe Ratio

LFM – Linear Frequency Modulation

LTI - Linear, Time-Invariant

MIMO – Multiple Input Multiple Output

MISL – Microwave Integrated Systems Laboratory of University of Birmingham

MR-MIMO - Minimum Redundancy MIMO

MTI - Moving Target Indicator

MUSIC - MUltiple SIgnal Classification

NI – National Instruments

PC – Personal Computer

PCB – Printed Circuit Board

PPS – Pulse per Second

PRF – Pulse Repetition Frequency

PRI – Pulse Repetition Interval

Radar – Radio Detection and Ranging

RCS – Radar Cross Section

RF – Radio Frequency

RMS – Root Mean Square

LIST OF ABBREVIATIONS

RVNS – Random Variable Neighbourhood Search

RX – Receiver

SA – Simulated Annealing

SAR – Synthetic Aperture Radar

SIMITAR - Persistent Surveillance from air with a low frequency MIMO Towed Array Radar

SNR – Signal to Noise Ratio

Sonar – Sonic Detection and Ranging

TDMA – Time Division Multiple Access

TX – Transmitter

USB – Universal Serial Bus

USRP - Universal Software Defined Radio Peripheral

UWB – Ultra Wideband

VNA – Vector Network Analyzer

VNS - Variable Neighbourhood Search

ABSTRACT

The aim of this research is to investigate a novel Multiple-Input-Multiple-Output (MIMO) sensor system for automotive applications. Compared to traditional phased arrays, a MIMO array can achieve the same fine angular resolution, but with a drastically less amount of sensor elements. For example, a MIMO array of 10 elements can deliver the same resolution as a phased array of 25 elements. The other highlight of this technology is that it can operate at short ranges, which is physically impossible with a phased array. Therefore, a MIMO system can potentially provide very high angular resolutions at short ranges. These properties make such a system attractive for a number of automotive applications, including parking aids, short-range cruise control, speed-over-ground estimation, pedestrian detection etc. Research started with the verification of application of conventional MIMO techniques for radar context. MIMO techniques were tested with the existing RF equipment in laboratory environment. Beamforming capabilities were verified, range and angular resolutions were compared to equivalent phased arrays and multiple-target resolving capabilities were confirmed. Nearfield focusing algorithms for MIMO arrays were developed and verified via experiments in the anechoic chamber with the same equipment. A technology demonstrator based on ultrasonic equipment was built and then tested in an anechoic chamber and the findings were compared to computed performance parameters. Further performance optimisations via aperiodic MIMO configurations were explored via use of heuristic optimisation algorithms. An optimised configuration then was tested in anechoic chamber and its performance was confirmed in experimentally. Finally using radio equipment again, an initial study on MTI applications was done. Platform motion compensation methods were developed and tested in order to make up for vehicular motion and to compensate for its possible effects. Both stationary platforms and a moving platform was used to experimentally confirm the MTI capabilities on both in an indoor setup and an outdoor setup.

1 INTRODUCTION

1.1 BACKGROUND

1.1.1 Radar/Sonar Concept

Radar is a sensing technology that employs electromagnetic waves. It stands for Radio Detection and Ranging. Similarly Sonar is a sensing technology that employs acoustic (sonic) waves for its purposes and it stands for Sound Navigation and Ranging. Although the physics of the two technologies are different the methods incorporated are inherently the same; that is using waves of an arbitrary medium and using their propagation and reflection characteristics to sense targets on their path.

A primary radar or a sonar measures the strength of the waves that reflect from a target, these waves could have been fired from the radar/sonar system (active) or they could be fired from any other transmitter (passive)[1]. What a radar does is to record reflected signals from a target area of interest. Then based on the recorded signals, a radar system can do a number of functions via signal processing; check for existence of targets, localisation of targets (range, crossrange, altitude, angle etc.) and similar. On top of these a final layer of functionality may be implemented to classify or even identify a target. In this thesis, the main interest is in finding a target's range and azimuth angle, and Doppler frequency if it's moving. The detection is usually the outcome of some signal processing, and associated with signal to interference (noise, clutter etc.) ratio to observe a peak in the return signal in time. Target range is usually implemented by measuring the fight time of the waves to reach the target and return to the receiver and multiplying it with the wave speed; speed of light in case of radar, and speed of sound in case of sonar (depending on if it's an underwater or not different speeds of sounds would be used for computing the flight time). An example of this kind of radar operation with collocated transmitter and receiver can be seen in Figure 1-1.

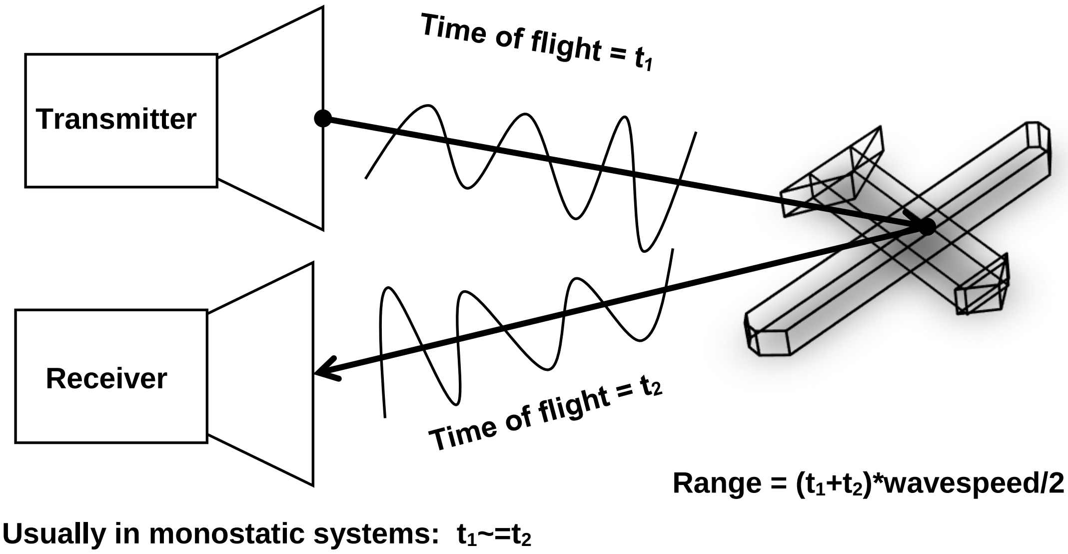


Figure 1-1 Basic radar/sonar operation and ranging assuming that the transmitter and receiver are collocated

In radar/sonar systems, one important parameter is the range resolution. Range resolution tells about the system's ability to be able to distinguish and resolve multiple targets in proximity. If two targets were to be closer than system's range resolution, than the return from these targets would show up as single target. This phenomenon is based on the transmitted waveform's properties; mainly on pulse duration and/or 3-dB bandwidth. When a simple non-modulated pulse is taken, range resolution is directly proportional to the pulse length, because for a non-modulated pulse the bandwidth is simply equal to inverse of the pulse duration as can be seen in equation 1-1[2]. This is because when a transmitted pulse travels in space and when it hits a target, it illuminates the target for the duration of the pulse width. If during this time, another target is to be illuminated with the same pulse, then the returns would add up causing one single strong return to be observed on the receiver within the same range bin. To achieve higher range-resolution –among other reasons-, modulated waveforms are often used. Modulated waveforms' range resolution is inversely proportional to their bandwidth. The way range resolution can be computed for modulated and non-modulated waveforms can be seen in equation 1-2[2].

$$Bw = \frac{1}{T} \qquad \textit{1-1}$$

$$\Delta R = \frac{c}{2 * Bw} \qquad \textit{1-2}$$

In equations 1-1 and 1-2; *Bw* stands for 3-dB bandwidth of the transmitted signal, *T* stands for pulse duration for non-modulated pulses, ΔR stands for range resolution, and *c* stands for speed of the travelling waves (e.g. speed of light, speed of sound in air, speed of sound in water etc.).

Two of the main components of a radar/sonar system are the transmitter/receivers or transducers. These components convert electrical signals into electromagnetic or acoustic waves that propagate over space or vice versa. These components usually have important properties that stand out among others, which are 3-dB beamwidth and gain. The 3-dB beamwidth of a transducer is defined as the steering angle where the power density is reduced by a factor of 2 or 3-dB. And the gain defines how much of the wave power is converted into electrical power (or the other way around) compared to an isotropic (non-directional) antenna or a transducer.

The 3-dB beamwidth of an antenna is usually related to its aperture size. The approximate 3-dB beamwidth of an aperture can be estimated as given in 1-3[3]. And the approximate 3-dB beamwidth of an aperture in radians can be estimated as given in 1-4[4].

$$HPBW_d = 51\frac{\lambda}{D} \tag{1-3}$$

$$HPBW_r = 0.88\frac{\lambda}{D} \tag{1-4}$$

In equations 1-3 and 1-4, $HPBW_d$ stands for half-power beamwidth in degrees, $HPBW_r$ stands for half-power beamwidth in radians, λ stands for wavelength, and *D* stands for the aperture length.

Early technologies of radar were incorporating mechanically rotating radars with narrow beamwidths. This would allow operators to estimate the angle of a target by linking the radar's heading in time to the target's reflection time. More advanced and current methods incorporate sensor arrays. The information collected from separate sensors can be used to localise target(s) in angle.

Like range resolution, another resolution parameter is the angular resolution. The criterion for angular resolution is usually the 3-dB beamwidth because like range resolution, it's a representative of maximum angle that targets needs to be apart in angular space to be resolved separately (i.e. if two targets fall into the same 3-dB azimuth bin, then they may show up as a single target).

Radar/sonar systems don't necessarily have to have the transmitters and receivers at the same locations. In fact they don't even have to include a transmitter at all. Regarding the transmitter and receiver configurations there are two main categories of sensing systems. These are monostatic and bistatic configurations. In a monostatic configuration –common configuration- the transmitter and receiver are collocated, or sometimes even they're the same transducer/antenna with a multiplexer/demultiplexer circuit. A bistatic configuration is where transmitter and receiver are separated by significant distance between them. More information on bistatic radars can be found in texts by Cherniakov[5] and Willis[6]. Some bistatic configurations don't even include a transmitter in the system but rather rely on existing transmitters of opportunity; an example is using navigation satellites as transmitters and using ground receivers for imaging purposes[7].

When a wave hits a moving target, the scattered wave usually changes frequency depending on the incidence angle and the speed of target. This phenomenon is called the Doppler shift. The change in frequency can be computed as given in 1-5[8]. This information can be then used to estimate various properties of a target; such as radial speed, target classification etc. It can also be used to overcome problems caused by clutter or foliage. GMTI radars also use Doppler frequency information to extract moving targets from clutter (moving or stationery), since those may have different Doppler frequencies [9].

$$f_{Doppler} = \frac{2 * v}{\lambda} \qquad \textit{1-5}$$

In equation 1-5, $f_{Doppler}$ stands for the Doppler frequency shift, v stands for the radial speed of the target, and λ stands for the wavelength.

1.1.2 Phased Sensor Arrays

Sensors can be placed in alignment with each other in specific configurations and can be used to obtain narrower beamwidths then an individual sensor can offer. In such configuration, when all the transmitters send out the same signal the signals are then combined in space forming a narrower beam. And often times with proper analogue circuitry or with proper signal processing in case of analogue to digital signal conversion, the transmitted or received beams can be steered electronically[10]. Array sensors can be in many shapes and in multiple dimensions. However, for the rest of this thesis the focus will be on 1-dimensional linear arrays. The newly obtained beam pattern is often referred to as the “Array Factor” and it can usually be derived by starting with the summation of transmitted or received signals. The process of using and manipulating the array factor is often referred to as “beamforming” and can be done either via analogue circuitry or via digital signal processing. Beamforming is done via shifting the phases of the received signals before obtaining the array factor sum, so that they would yield a maximum SNR if a prerequisite(i.e. target in a specific angle) is fulfilled[11]. The array factor is usually used in conjunction with the sensor element pattern to draw a final beam pattern. This final pattern is the multiplication of the array factor and the element pattern[12]. Often times though, element pattern is ignored since it makes the computation of array beam pattern easier, the element pattern is considered as merely additional weighting on the final beam pattern and usually associated with limits on field-of-vision.

The structure of a uniform linear array looking at a target at far field can be generalized as below for our purposes (Figure 1-2). Due to the far-field assumption the paths running from target to transceivers are assumed to run parallel as shown in the figure. The array elements and the target(s) are all assumed to be on the same plane (XY plane). Phase centre of the array is taken as the physical centre of the array to cover the cases of both odd and even numbered arrays. All lengths discussed are electrical lengths to properly allow wavelength scaling or carrier medium changes.

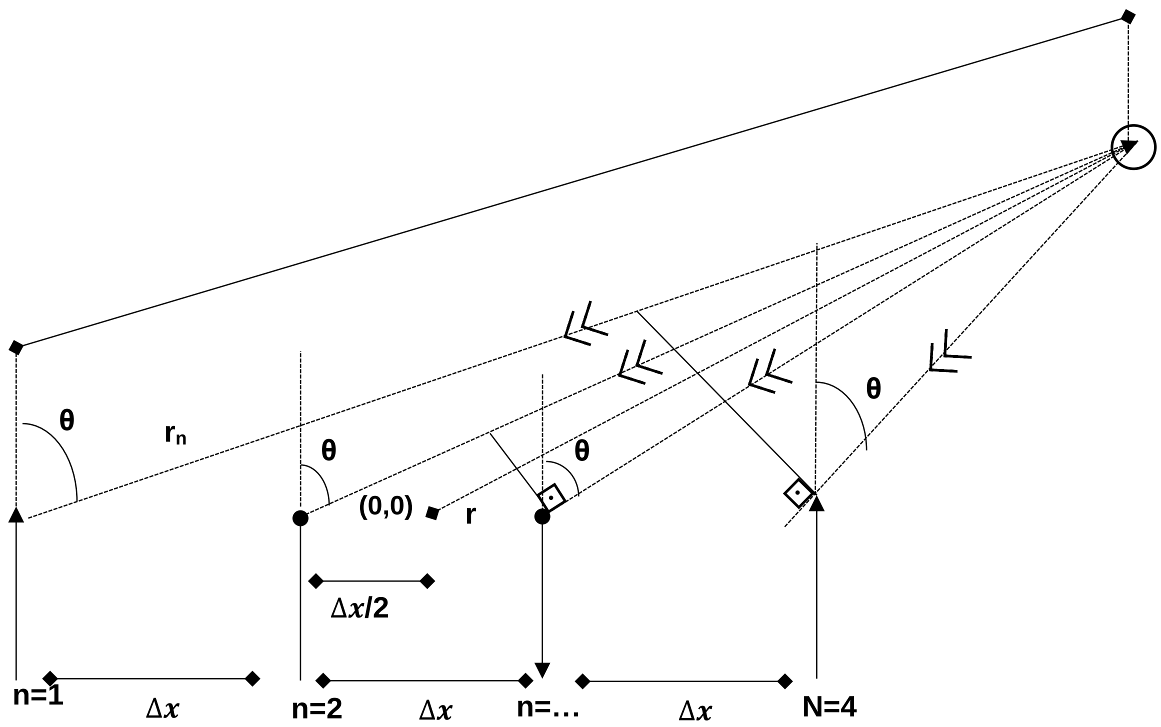


Figure 1-2 Generic uniform linear array geometry

In Figure 1-2 above, `$n$` represents an arbitrary array element number where N represents total number of array elements (so n can take values between 1 and N inclusive). Δx represents the uniform distance between each array element in electrical lengths (in terms of wavelength, denoted as λ further on). Θ represents the azimuth angle from the centre of the array to an arbitrary target and due to approximations it stands for specific azimuth angles from corresponding array elements to target. Similarly `$r$` stands for the range from the centre of the array to an arbitrary target, whereas r_n stand for a specific range from a corresponding array element.

Assuming the two-dimensional geometry in Figure 1-2, we can have the above assumption given in 1-6 at far field [13], which is the assumption that target range is considerably larger than the Fraunhofer distance. This assumption allows us to approximate all the incident angles from a target to sensors to be the same.

$$r \gg \frac{(N\Delta x)^2}{\lambda} \qquad \text{1-6}$$

The same approximation then allows us to draw perpendicular lines between incident lines and re-write the sensor-to-target ranges so that the distances from transmit and receive elements to a target r_n can be written as 1-7, where r is the range from the centre of the array to the target, θ is the angle from the centre of the array to the target, n is the array element number, Δx is the sensor element spacing, N is the total number of sensors. These equations tell that the phase difference between each element is basically $\Delta x * sin(\theta)$ for the array.

$$r_n = r + \left(\frac{N-1}{2} - (n-1)\right)\Delta x * sin(\theta) \qquad 1\text{-}7$$

This path notation can be used to calculate the phase difference of a signal which travels from a transmitter to a target and then back to a receiver by using complex notation e^{-jkr_n}. Then array factor of any array which is the sum of signal phase differences can be written as in 1-8 [14].

$$AF = \frac{1}{N}\sum_{n=1}^{N} e^{-jkr_n} \qquad 1\text{-}8$$

In equation 1-8 AF stands for array factor, and k stands for wavenumber which is obtained by $k = 2\pi/\lambda$, the other variables are as before. Substituting 1-7 in 1-8, for a uniform linear array due to uniform and linear progression of the phase at far field array factor can now be rewritten as in 1-9.

$$AF(\theta) = \frac{1}{N}(N * e^{-jkr})\sum_{n=1}^{N} e^{-jk\left(\frac{N-1}{2}-(n-1)\right)\Delta x sin(\theta)} \qquad 1\text{-}9$$

When absolute value of array factor is taken the array factor simplifies and approximates to a sinc function with respect to θ as shown in 1-10 [13].

$$|AF(\theta)| = \left|\sum_{n=1}^{N} e^{-jk\left(\frac{N-1}{2}-(n-1)\right)\Delta x sin(\theta)}\right| \approx sinc\left(\frac{\mathrm{N}\sin(\theta)\,\Delta x}{2}\right) \qquad 1\text{-}10$$

And we know that the array can virtually steered by applying complex weights to the array factor as shown in 1-11 where steering weights can be computed as also shown in 1-12 [14]. This now yields a controllable (steerable) narrow beam looking at a specific direction. This kind of electronic steering of a beam however results in beam broadening [15]. The electronic scanning effectively reduces the aperture size looking at the target relative to the incident angle. Therefore the beam broadens by a factor of $\cos(\theta_{steer})$, and so the beamwidth estimation given in 1-4 becomes as given in 1-13.

$$AF(\theta) = \frac{1}{N}\sum_{n=1}^{N} w_n(\theta) * e^{-jkr_n} \qquad \text{1-11}$$

$$w_n(\theta) = e^{jksin(\theta)\left((n-1)-\frac{N-1}{2}\right)\Delta x} \qquad \text{1-12}$$

$$Bw_r = 0.88\frac{\lambda}{D * \cos(\theta_{steer})} \qquad \text{1-13}$$

In equation 1-13, λ stands for wavelength, D stands for aperture length, θ_{steer} stands for the steering angle. Further inspection of array factor for a uniform linear reveals that in order to avoid having grating lobes, the element spacing should be equal or less than half-wavelength[13]. Array factor computed using this method for a sensor array with 16 elements and half wavelength spacing can be seen in Figure 1-3.

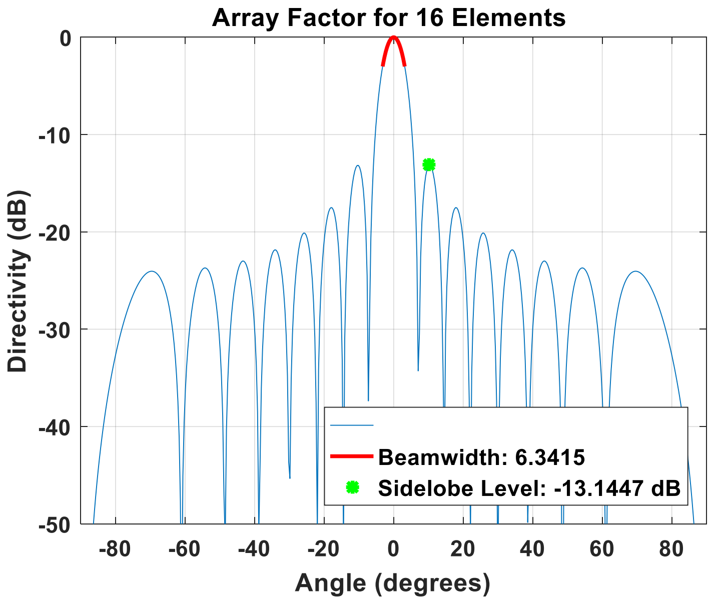


Figure 1-3 Array factor for a sensor array with 16 elements with half wavelength spacing

1.1.3 Phased Arrays in Near-field

The aforementioned array factor calculations and simulations only work however when the target is significantly far away from the radar so that the approximations would hold; that is the incident angle to each sensor would be numerically similar. When the intended scope of the array is near-field as in ranges that are close to or less than the aperture's Fraunhofer distance, these approximations and the beamforming functions are almost guaranteed not to work. However, there are methods in the literature to counteract the effects of being in the nearfield. Most of these methods employ digital beamforming on receive and/or transmit. One of the examples that don't rely on digital or analogue beamforming techniques, proposes adjusting the shape of one of the array, literally transforming the linear array into a lens-shaped array to be able to focus in near-field, and further explains the process of moving one of the three elements in order to be able to adjust the focusing range [16]. Digital or analogue beamforming techniques used for near-field focusing can actually be seen as the same operation applied to signals only in digital space. One of these methods propose applying phase shifts to received signals of a phased array to simply adjust the obtained beam pattern into a desirable one[17]. The same authors then take this approach and use ideal far-field phases to compensate the near-field phase shifts and convert any obtained pattern at any range into a far-field beam pattern [18]. More information on how these methods operate will be given in the next chapters and will be used to develop MIMO near-field focusing techniques. Another research proposes applying the phase shifts – also referred as fractional delays in the paper- in the form of time-domain FIR filters before the beamforming happens [19].

1.1.4 Thinned Phased Arrays

Sometimes for large arrays with half-wavelength spacing, the number of required elements for producing an array for a needed beamwidth might be unacceptable due to various reasons (weight, cost, lack of computation power etc.). In such cases the filled array can be thinned by removing non-edge sensor elements (i.e. without reducing the total aperture size) [14]. This usually comes in with a trade-off from gain and possible other trade-offs within the beam pattern (such as the sidelobe-levels, locations of the nulls and sidelobe etc.). However, if the array is large enough and/or if thinning is done properly this method can lead to significant optimisations. Currently, the most common method to thin phased arrays is to simply randomise which elements to reduce from the array. Other methods involve using heuristic algorithms to find optimisations in a more intelligent manner, such as using genetic algorithm[20] or simulated annealing[21]. Heuristic algorithms work to find a good solution to a given problem without doing an exhaustive search and therefore they don't scan the entire solution space. An example of such a thinned array with the aperture size of 16 elements with half wavelength spacing but with 13 elements can be seen in Figure 1-4. The 4th, 9th and 12th elements were removed randomly to obtain this pattern. Note that this is basically the same half-wavelength spaced array as before with 3 elements randomly removed. Compared to its filled counterpart in Figure 1-3 this configuration seems to offer a smaller beamwidth of 6 degrees but with raised overall sidelobe levels (i.e. higher integrated sidelobe level ratio – ISLR).

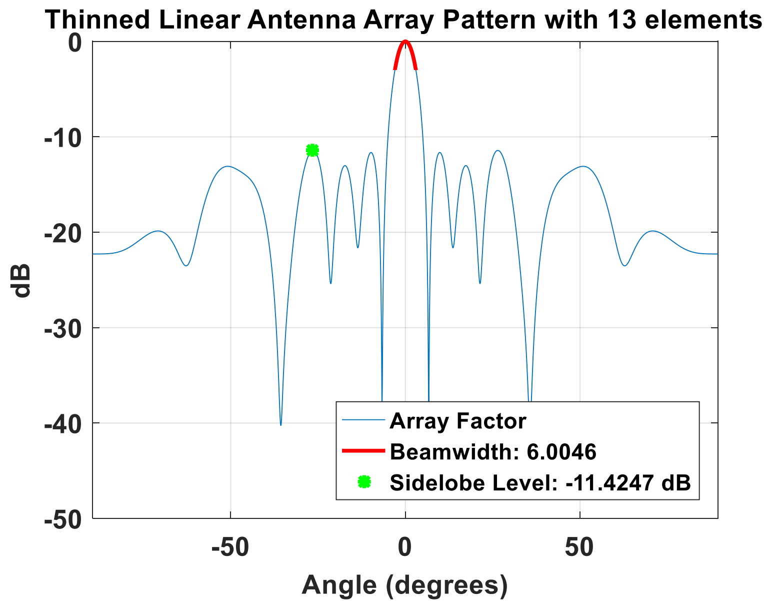


Figure 1-4 Thinned linear array factor with 13 elements with the aperture size of a 16 element array

1.1.5 **Matched Filtering**

A matched filter, by definition, is a special type of filter designed to maximise the SNR of a given signal in the presence of noise typically.

The impulse response of such a filter is the reverse conjugate of the original waveform as given in 1-14, where *h* is the impulse response, *alpha* is an arbitrary constant, *x* is the original waveform, T_m is the time when the SNR will be maximised [22]. As such, the convolution of the signal with its reverse conjugate yields the maximum possible output if computed as shown in 1-15.

$$h(t) = \alpha x^*(T_m - t) \tag{1-14}$$

$$y(t) = \int_{-\infty}^{\infty} s(t) * \alpha x^*(T_m - t) dt \tag{1-15}$$

In equation 1-14, *h(t)* is the time domain impulse response of the matched filter, α is a constant multiplier, *x(t)* is the input function or the transmit signal for our case, T_m is usually the pulse length or the integration time, and *t* is the time variable. In equation 1-15, *y(t)* is the output of the matched filter, and *s(t)* is the received signal.

Matched filtering has two important roles in the scope of this research. First role is range compression, which allows the transmission of longer frequency modulated pulses without losing range resolution as well as maximising SNR. Second role is the signal maximisation and separation; the entire MIMO theory rests on signal processing being able to separate transmitted signals from each other[23]. With proper waveform design, which is maximum auto-correlation and minimal cross-correlation between transmitted waveforms, matched filtering can effectively separate the signals that were summed in space. This was paid less attention in this research since TDMA scheme was mainly used for signal orthogonality.

However, in signal processing convolving time-domain signals is a relatively computing heavy work. A less computing heavy approach to the problem of matched filtering multiple –received- signals to multiple –transmitted- signals is to first take the signals into frequency domain with Fourier transformation [24].

Then the matched filter output can be computed as simple as a vector multiplication in frequency domain as shown in 1-16, where *Y* is the Fourier transform of the matched filter output, and *S* is the Fourier transform of the received signal. If needed, the matched filter output in frequency domain (*Y*) can then be brought back to time domain for range/time analysis.

$$Y(f) = S(f) * F\{\alpha x^*(T_m - t)\} \qquad \textit{1-16}$$

In 1-16, *Y(f)* is the frequency domain matched filter output (i.e. Fourier transform of *y(t)*), *S(f)* is the received signal in frequency domain (i.e. Fourier transform of *s(t)*), and *F{…}* stands for the Fourier transform on a function, rest of the variables are as was before. One other way to summarise the matched filter operation is that it is the cross-correlation of the received signal and the reference (transmitted) signal.

1.1.6 **MIMO Array Concept**

A more comprehensive insight into MIMO arrays will be given in Chapter 3, however the concept itself is briefly mentioned in this section. A Multiple-input-multiple-output sensor array is by definition only a sensing system that consists of multiple transmitters and multiple receivers. However, the MIMO array in the context of this thesis is specifically a 1D collocated, pulse coherent MIMO sensing array. 2D arrays were briefly studied to have a better understanding of the virtual array concept, however they are not within the scope of this research. The basis of MIMO array relies on transmission of orthogonal signals. The orthogonality scheme could be any of the schemes, such as FDMA, CDMA, TDMA or even others like with spatial orthogonality. In a collocated MIMO array some of the sensors are usually designated as receivers and others are designated as transmitters, so it can be said that it's a combination of a transmit sub-array and a receive sub-array. There are variations of MIMO arrays which utilise sensors as both transmit and receive elements, however this is beyond the scope of this research.

Fundamental theory of MIMO is that the orthogonally transmitted waveforms from separate transmitters can then be extracted in reception via signal processing which then can be used for separate transmit and receive beamforming [23]. This means that beamforming is done at the signal processing level, and the beam does not physically combine in space if signals are truly orthogonal. And this results in creation of a "virtual array" (or sometimes referred to as "virtual aperture"), which usually has the beam pattern (MIMO array factor) as the multiplication of receive array factor and transmit array factor [25]. Compared to traditional phased arrays, a MIMO array can achieve the same fine angular resolution, but with a drastically reduced amount of sensor elements. This means there is loss in antenna gain, but due to persistent illumination (like a spotlight radar), integration time can be increased in order to make up for the lost gain, but with the added advantage persistent operation. For example, a MIMO array of 10 elements consisting of 5 transmit and 5 receive elements can deliver the same resolution as a phased array of 25 elements (i.e. a 5x5 MIMO array). The other highlight of this technology is that it can operate at short ranges, which is physically impossible with a phased array as the beam requires significant distance from the antenna aperture to form. Therefore a MIMO system can potentially provide very high angular resolution at short ranges.

A MIMO array uses the advantage of orthogonality -in time, frequency or code- for various advantages [26]. Here, the concept of orthogonality means low correlation of transmit waveforms from transmit antennas. These advantages can be; reduced amount of necessary array elements, increased angular resolution compared to same number of array elements, multiple target tracking, simultaneous search

& track etc. Depending on the application these advantages are available to use via digital beamforming. This technique simply means that the beamforming process does not happen in the electromagnetic space but rather in the digital space. And this is only achievable due to the low correlation of the transmitted signals. Because only then the transmitted signals can be distinguished at the receivers and can be processed coherently according to one's needs. A detailed explanation of how a MIMO sensor array works is given in the next chapter with supporting equations, figures and explanations.

1.1.7 Near-field MIMO Arrays

The current state of the near-field MIMO arrays is still mostly at a theoretical level, however there is still some research regarding the subject with experimental work. Researchers from China has implemented and tested a 2D MIMO array using ultra-wideband near-field focusing techniques [27]. Another group of researchers from Germany also had experimental success with 2D near-field MIMO imaging with similar methods [28]. Both these research are based on the research done by Zhuge and Yaroyov about 3D MIMO near-field imaging using range-migration and image reconstruction techniques [29]. Zhuge and Yaroyov's research is based on initial research on method of near-field image reconstruction in another paper -unrelated to MIMO- by Yaroyov et al. [30]. All these methods however seem to rely on UWB, CW or even SAR methods. Another research that presents simulated results for a proposed MIMO array system for possible substitution of any systems requiring high number of elements uses the Backprojection algorithm instead of conventional beamforming techniques [31]; this method surpasses the range limitations introduced by array systems but introduces higher computation load and system complexity. Similarly, a research from Germany summarises some considerations to take regarding a short-range MIMO system; in their simulations they also use Backprojection algorithm for near-field imaging [32]. Research from Duke University proposes and demonstrates the use of monostatic MIMO arrays for increased scan space coverage [33]. Note that monostatic MIMO here refers to using same sensors (antennas) both as transmit elements and receive elements, but not in the sense of the separation of transmit and receive arrays. And due to this fact, design of the virtual array has less freedom of axis. And so, their use of MIMO methods and the reasons behind it are not alignment with the scope of this research. Finally there is research published by NXP

Semiconductors engineers, which describes an experimental near-field 2D MIMO imaging system with Doppler processing using FMCW[34]. And even though most of the details including beamforming method and orthogonality scheme are omitted, the near-field problem seem to be addressed by using the 79GHz band to drastically reduce the Fraunhofer distance and put possible targets in the far-field of the implemented array.

1.1.8 Thinned MIMO Arrays

Like the near-field MIMO array subject thinned MIMO arrays subject is also mostly at a theoretical level with little experimental results. In fact, the mere use of MIMO arrays is considered as a thinning technique by some researchers [35]. Nevertheless, the idea of optimising MIMO sensor arrays go as back as 2008; a research that explains the idea of minimum-redundancy MIMO arrays [36], which is based on the minimum-redundancy linear arrays [37]. And there are quite a few examples in literature that addresses the design of such arrays. One research proposes the use of combinatorial methods of difference bases and cyclic difference sets to design minimum-redundancy MIMO arrays [38]. Another research from China proposes a two-step construction method for MR-MIMO arrays, which is figuring out the minimum number of necessary elements for a given aperture size first and then expanding the virtual array aperture [39]. Another research from Germany resorts to using an optimisation method called Particle Swarm Optimisation for designing an MR-MIMO a [40]. Also from Germany, researchers from Munich Technical University have developed and introduced their own algorithm for designing minimum-redundancy MIMO arrays [41]. Other researchers introduce a kind of deterministic thinning method, constructing MR-MIMO arrays by using cyclic permutation of perfect distance method, as an alternative to brute-force or stochastic optimisation methods [42]. Research led by similar group of engineers also propose the use of cyclic difference sets like others and a stochastic method like Simulated Annealing to address the design of MR-MIMO arrays [43]. Researchers also from Central South University in China had success with using almost difference sets (ADSs) for thinning MIMO sensor arrays [44]. Another uses that utilises a stochastic approach for the design of a thinned array uses genetic algorithm to introduce distance perturbations to transmit and receive arrays synchronously [45]. Similarly, there is another research that uses Simulated Annealing to design a sparse virtual array as the output of MIMO array [46]. A minor difference in this research is that this research was aimed to minimise the number of necessary matched filters for a sonar application rather than a radar. However, unfortunately none of these research provide experimental results but mostly simulated results. As far as the experimental results with thinned and/or minimum-redundancy MIMO arrays go, there are some significant experimental results obtained by researchers from Institut für Hochfrequenztechnik from Techn. Univ. Hamburg-Harburg. Researchers Rezer and Jacob have successfully thinned a monostatic linear MIMO array with 4 transceivers and experimentally verified its capabilities [47]. Also from the same institute, there is another research with experimental results of a thinned MIMO frame array for 3D imaging [48].

1.1 PROBLEM STATEMENT

For automotive systems, surveillance and situational awareness has always been an important subject. Often times, the observation of surroundings is the task of the driver. The problem with that is driver's cognitive abilities are limited; either by his/her physical abilities like impaired vision, limited view angle or environmental conditions such as lack of light, weather conditions etc. Not being able to see or detect surrounding objects is a major cause for collisions where lives might be endangered. Also with the recent rise of self-driving or assisted-driving technologies, situational awareness is even more important than ever.

The problem can be summarized as; the lack of ability to detect and range the objects in short/medium proximity to an automobile with high resolution with feasible costs. More specifically, what's needed is a

- sensor array with scanning capabilities,
- for short range,
- and for situational awareness in a vehicular environment.

This sensor array should be able to

- locate targets in range and azimuth with a fine angular resolution

- with as little as possible amount of sensors,
- for low cost and low profile production (cost-effective),
- with the ability to detect moving targets as opposed to stationary targets.

Note that, because the scope of this study is a generic concept, some of the concepts have been proven with radio frequency equipment whereas some others have been proven with acoustic sensing equipment. This research in the end is an experiment driven research and practical results were aimed and obtained. Therefore, even though MIMO sensor arrays for this problem is a good approach on its own, further studies were needed for near-field operation and for further improvements.

1.1.1 Thesis Contributions

There are many aspects to the final proposed solution, some of them old and well known, some of them completely new. And the research output is a mixture of these technologies.

The phased array systems have been around for a while. Phased arrays are –in simplest terms- linearly placed transceivers which are able to electronically scan an area. They have hard physical limitations on closest range they can scan and how much angle they can scan. One way to overcome this problem is to digitally receive the data and apply signal processing algorithms to focus the image. The idea of phased arrays with short-range processing is not new[17], but to apply this idea to MIMO arrays and even to aperiodic MIMO arrays are novel concepts. There are also other algorithms used to solve the problems with increased system complexity; such as time aligning received signals, phase corrections, aperiodic beamforming and various other speed oriented computation optimizations. Short-range processing for MIMO arrays and aperiodic MIMO beamforming -along with other supporting digital signal processing algorithms- are developed by us and there is no match of it in the literature.

Separately, phased arrays can be widened to achieve higher resolution with the same number of elements by trading-off from other performance aspects (eg. maximum scan angle, sidelobe levels)[12, p. 92]. The idea to apply these methods to MIMO arrays to obtain aperiodic MIMO arrays, and to control the trade-off is currently only at a cutting-edge research level with limited number of examples in the literature as explained before. We used three of the most popular global optimization algorithms to come up with a special configuration that can fulfil required specifications

Overall, the final solution is a combined technology of aperiodic MIMO arrays specialized for short-range scanning developed with global optimization methods along with other supporting signal processing algorithms.

In the literature there is no other example of an experimental aperiodic MIMO array for short-range applications, let alone any products or even prototypes. There are also no other solutions out there that can give the same high performance our solution can give without increasing the production and manufacturing costs significantly.

1.2 METHOD (THESIS OUTLINE)

The method of this project is also the outline of this thesis. The project has been set up to follow a linear progression and development, which is also the best way to break down and put across the details of this project. Below is a list of chapters of this thesis and short explanations of their content. Chapter 1, the “Introduction” chapter gives background information on radars, phased arrays, MIMO arrays and thinned MIMO arrays. And then explains the problem, proposed solution and method (thesis outline). Chapter 2, the “1D MIMO Sensor Array in Far-field” chapter derives the MIMO array factor. And then proves the MIMO sensing theory experimentally with supporting simulations with existing lab equipment. Chapter 3, the “1D MIMO Sensor Array in Near-field” chapter derives the MIMO array factor in near-field this time. And then explains the development of an algorithm for focusing the signals in near-field. After that, tells about the testing and verification of the algorithm with supporting simulations with existing lab equipment. Chapter 4, “Experimental System Design & Results” chapter explains the design and implementation of a purpose specific and modular technology demonstrator. And then tells about the testing and verification of the demonstrator is then experimentally and with supporting simulations. Chapter 5, “MIMO Sensor Array Optimisation & Results” chapter implements heuristic optimisation algorithms to further optimise the performance of MIMO sensor array, namely the aperiodic MIMO arrays. It then explains the experimental testing and verification of a chosen aperiodic MIMO configuration. Finally it presents results of a measurement of a practical object with the aperiodic MIMO technology demonstrator. Chapter 6, “Conclusions and Future Work” summarises the conclusions of this research has given, it gives pointers to what the next steps regarding this research specifically and the research field itself.

2 MIMO SENSOR ARRAY IN FAR-FIELD

The process of beamforming via analogue or digital techniques in uniform linear phased arrays was discussed in Chapter 1. Beamforming and beam steering techniques for MIMO arrays are considerably different to than phased array systems due to basic principal differences in concepts. As previously mentioned in Chapter 1, the major differences are 1) the separation of transmit and receive arrays, 2) the element spacing in transmit and receive arrays and 3) the transmission of "orthogonal" waveforms by each transmit element. In this chapter the array factor of a MIMO sensor array will be derived with a similar method to a phased array. The element phase differences for an arbitrary MIMO sensor array will be analytically derived at far-field. Finally, application of these techniques will be presented via experiments and compared to corresponding simulations, to confirm understanding of these basic principles and as a starting point for the research objectives set out in this thesis.

2.1 INTRODUCTION

In this chapter, the derivation of MIMO array factor in far-field, how an experimental setup was built, how a simulation programme to accompany experiments was written, and the results reached via experiments and their corresponding simulations are explained. First section introduces a simple MIMO radar geometry in far-field and derives the MIMO array factor mathematically using equations and system block diagrams. Later on, the system block diagram of an experimental setup is shown and explained. Then, initial simulations consisting of far field scenarios are shown with the simulation parameters and compared with the computed beam patterns using the equations from previous sections. Afterwards, experimental results with spherical targets in an anechoic chamber are presented with their corresponding simulations. Finally, conclusions are presented to move on to next chapter where we discuss MIMO sensor array in near-field.

2.2 MIMO Array Factor Derivation

It turns out that, a linear MIMO radar's array-factor can be rewritten as the multiplication of transmit and receive array factors. The reason behind this phenomenon is because, the orthogonally transmitted waves can be separated digitally after receiving.

Orthogonality of signals can be provided with one of the multiple access schemes; frequency division (FDMA), code division (CDMA) and time division (TDMA). Performance of different multiple access schemes in MIMO applications is still an active research area[49]–[52]. But it's known that, only TDMA offers zero cross-correlation in a pulse radar system if the time gap between transmit sequences is left long enough, and that not having some cross-correlation can impact system performance such as by causing higher sidelobe levels. For this reason TDMA was considered for the remainder of this thesis.

For the clarification of variables and signals a sketch of a 2x2 MIMO array looking at a target at far field has been given in Figure 2-1. A 2x2 MIMO array means it has a 2 element transmit subarray and a 2 element receive subarray. Due to the far field assumption the paths running from the target to transmitters and receivers are assumed to be parallel as shown in the figure. Note that the sketched MIMO array is a special case where targets are assumed to be in the far-field of both transmit and receive arrays. Therefore all azimuth angles to a target are assumed the same and, therefore all the ranges from the linear MIMO array elements are assumed to be running parallel, as shown in Figure 2-1.

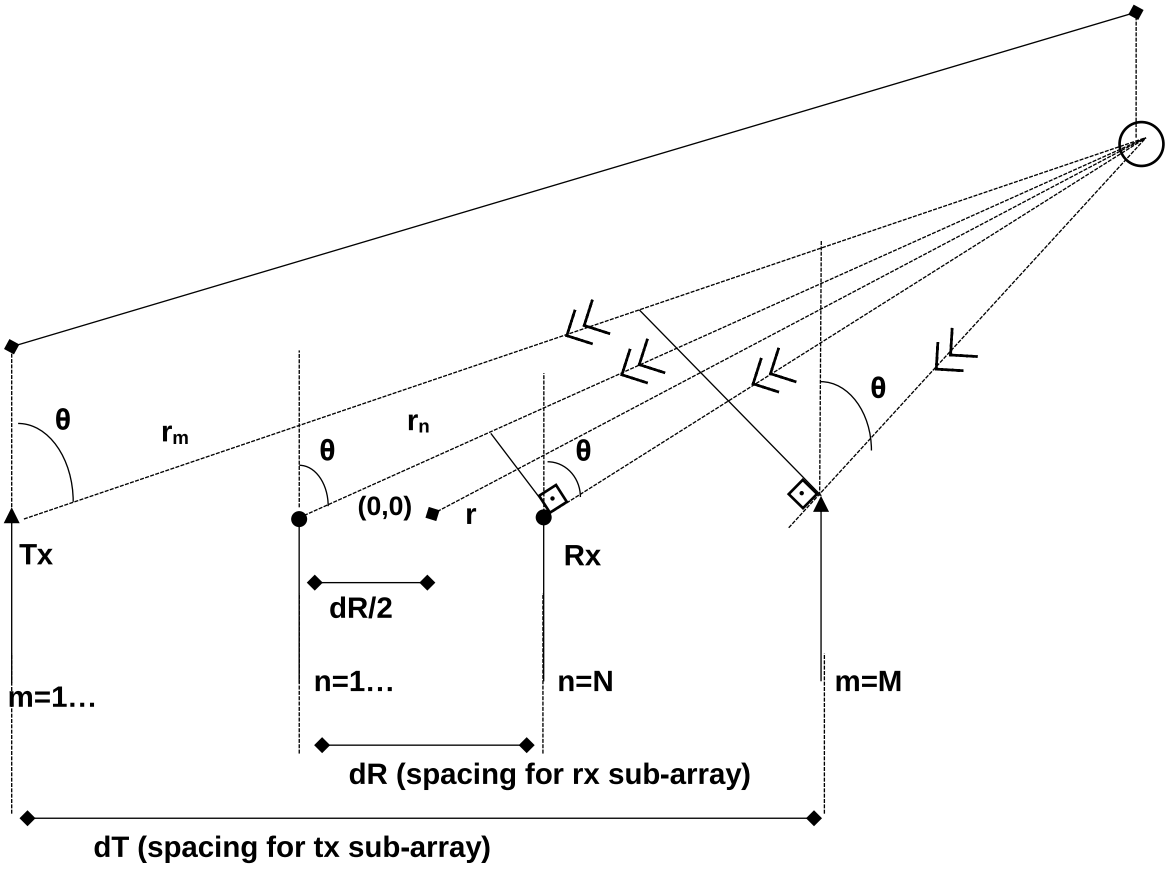


Figure 2-1 2x2 MIMO array sketch with far-field approximations

$$r_n = r + \left(\frac{N-1}{2} - (n-1)\right) dR * sin(\theta) \tag{2-1}$$

$$r_m = r + \left(\frac{M-1}{2} - (m-1)\right) dT * sin(\theta) \tag{2-2}$$

The distances from transmit and receive elements to a target r_m and r_n can be written as 1-7 and 2-2 respectively using the far-field approximations where r is the range from the centre of the array to the target, θ is the angle from the centre of the array to the target, m is the transmit element number, n is the receive element number dR is the receive element spacing, dT is the transmit element spacing, M is the total number of transmit antennas, N is the total number of receive antennas. These equations show that the path difference between each element is basically $dT * sin(\theta)$ for transmit array and $dR * sin(\theta)$ for the receive array.

$$s_{mt}(t) = s_m(t) * e^{-jkr_m} \tag{2-3}$$

Signal $s_{mt}(t)$ from the m^{th} transmit element at an arbitrary point target can be written as in equation 2-3, where k is the wavenumber $\frac{2\pi}{\lambda}$ and λ is the wavelength, r_m is the range as in 2-2, and $s_m(t)$ is the signal transmitted from the m^{th} transmit antenna.

$$s_t(t) = \sum_{m=1}^{M} s_{mt}(t) = \sum_{m=1}^{M} s_m(t)e^{-jkr_m} \tag{2-4}$$

$$s_t(t) = \sum_{m=1}^{M} s_{mf}(t)e^{-jkr_m} = s_{mf}(t) * \sum_{m=1}^{M} e^{-jkr_m} \tag{2-5}$$

The illumination on the target by all transmit waves $s_t(t)$ can then be written as the sum of these signals as above in 2-4. Normally, if the signals $s_m(t)$ were to be non-orthogonal this sum would be impossible to separate because this sum happens in space. However due to the assumed (designed) orthogonality of the transmit waveforms, summation of $s_m(t)$ signals can be separated via matched filtering. After matched filtering $s_m(t)$ becomes $s_{mf}(t)$; correlation of $s_n(t)$ with $s_m(t)$, where m is the transmit element number and n is the receive element number again. By waveform design or by transmission scheme the signals $s_{mn}(t)$ should be similar so that they can be summed constructively, this similar signal is denoted as $s_{mf}(t)$, which is independent of m and n, therefore it can be moved outside of the summation as above in 2-5.

$$s_n(t) = s_t(t) * e^{-jkr_n} \tag{2-6}$$

$$s_n(t) = \left(s_{mf}(t) \sum_{m=1}^{M} e^{-jkr_m} \right) * e^{-jkr_n} \tag{2-7}$$

$$s_r(t) = \sum_{n=1}^{N} \left(\mathrm{s_{mf}(t)} \left(\sum_{\mathrm{m}=1}^{\mathrm{M}} \mathrm{e}^{-\mathrm{jkr_m}} \right) * \mathrm{e}^{-\mathrm{jkr_n}} \right) \tag{2-8}$$

$$s_r(t) = \mathrm{s_{mf}(t)} * \sum_{n=1}^{N} \left(\left(\sum_{\mathrm{m}=1}^{\mathrm{M}} \mathrm{e}^{-\mathrm{jkr_m}} \right) * \mathrm{e}^{-\mathrm{jkr_n}} \right) \tag{2-9}$$

The received signal $s_n(t)$ from an arbitrary point target then can be written as above in 2-6. Which can then be substituted with 2-5 as in 2-7. And then the sum of all received signals $s_r(t)$ can be written as in 2-8. Matched filter output signal $s_{mf}(t)$ is also independent of the outer summation, it can be moved outside of this summation too as shown above in 2-9. A MIMO array's matched filtered output is equivalent to a virtual array where its element positions are convolved. The matched filtered signals can be interpreted as the signals out of these virtual array elements (Figure 2-2).

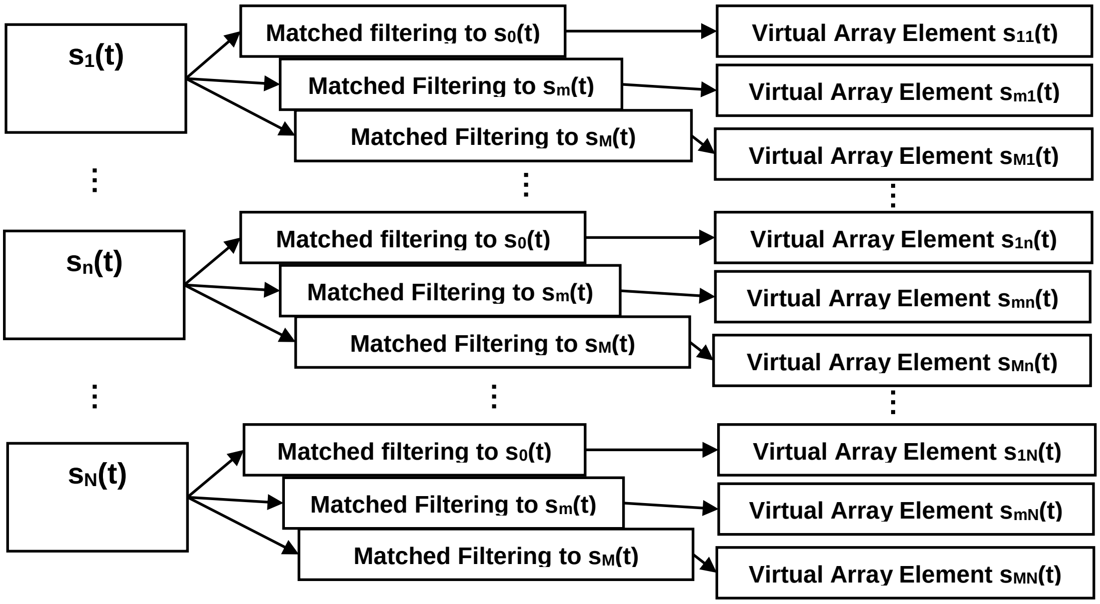


Figure 2-2 Matched filtering of received signals to extract virtual array element signals

$$s_r(t) = s_{mf}(t) * \sum_{n=1}^{N} e^{-jkr_n} \sum_{m=1}^{M} e^{-jkr_m} \tag{2-10}$$

Equation 2-9 implies that there M times N combinations of $s_r(t)$, which also explains how the virtual array is formed. The elements of the virtual arrays are the different possible $s_r(t)$ signals that are constructed by different m and n combinations. So each virtual element would receive a signal which travels the path of $r_n + r_m$ (from transmitter to target and then to receiver). Equation 2-9 also means that the virtual array element positions are the convolution of transmit and array element positions. An example of this convolution can be seen in Figure 2-3.

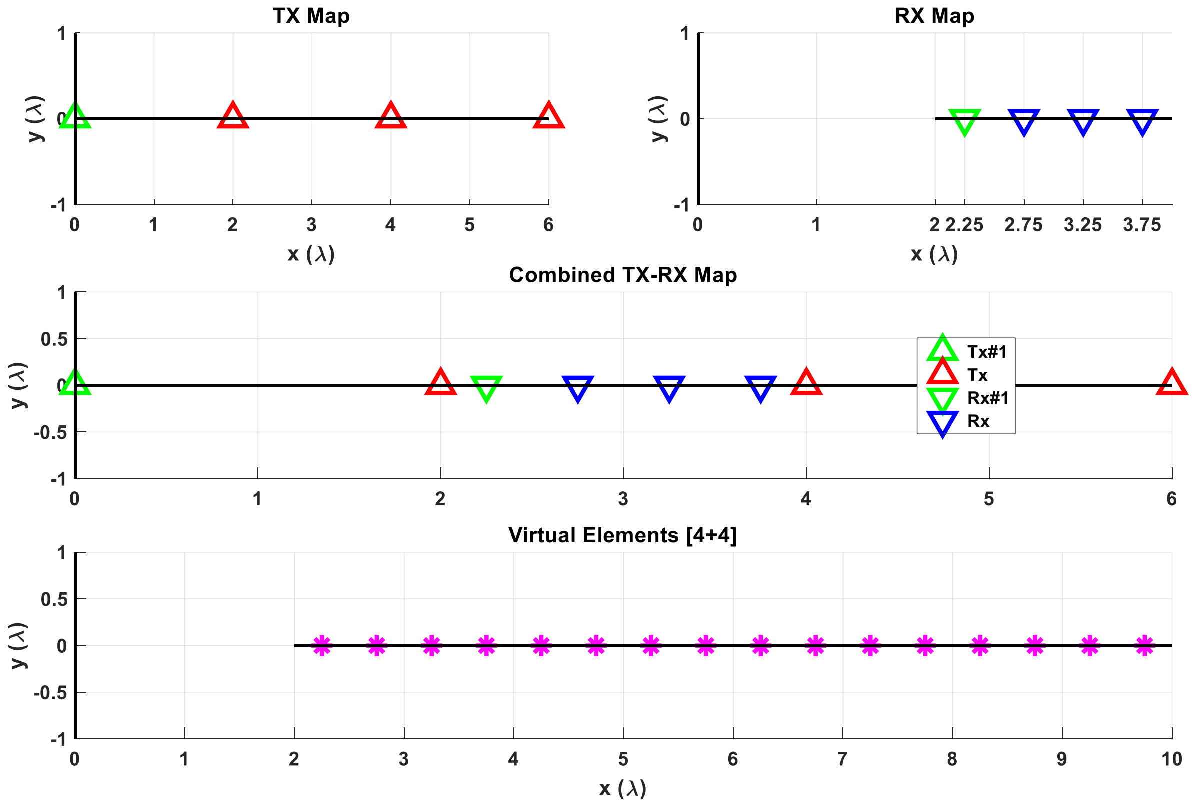


Figure 2-3 A MIMO array configuration map showing the convolution of a 4x4 MIMO array and its virtual array

Continuing the derivation; since the inner summation is constant with respect to the outer summation, inner summation in 2-9 can be moved outside as in 2-10. To be able to split this summation in two separate sums, r_m must be independent of *n*, as shown in 2-2.

$$AF_{mimo}(\theta) = \sum_{n=1}^{N} e^{-jkr_n} \sum_{m=1}^{M} e^{-jkr_m} \quad \text{2-11}$$

$$AF_{mimo}(\theta) = AF_{receive}(\theta) * AF_{transmit}(\theta) \quad \text{2-12}$$

And from 2-10 the MIMO array factor can be extracted as in 2-11, because each summation is the array factor of the corresponding antenna array [53, p. 327]. And so the MIMO array factor can then be presented as the multiplication of transmit and receive array factors as shown in 2-12. Knowing this, a coarse sub-array can be used with a fine sub-array to cancel out grating lobes of the coarse array with the fine array's nulls. This results in the conventional MIMO array design given in 2-13[23].

$$dR = \frac{\lambda}{2}, \qquad dT = N * dR \quad \text{2-13}$$

Where dR is receive element spacing, λ is wavelength, dT is transmit element spacing, and N is the number of receive elements which has to be a positive integer. In Figure 2-4 MIMO array pattern and individual sub-array patterns of a 4x4 MIMO array can be seen and the matching of the nulls to grating lobes can be seen in action.

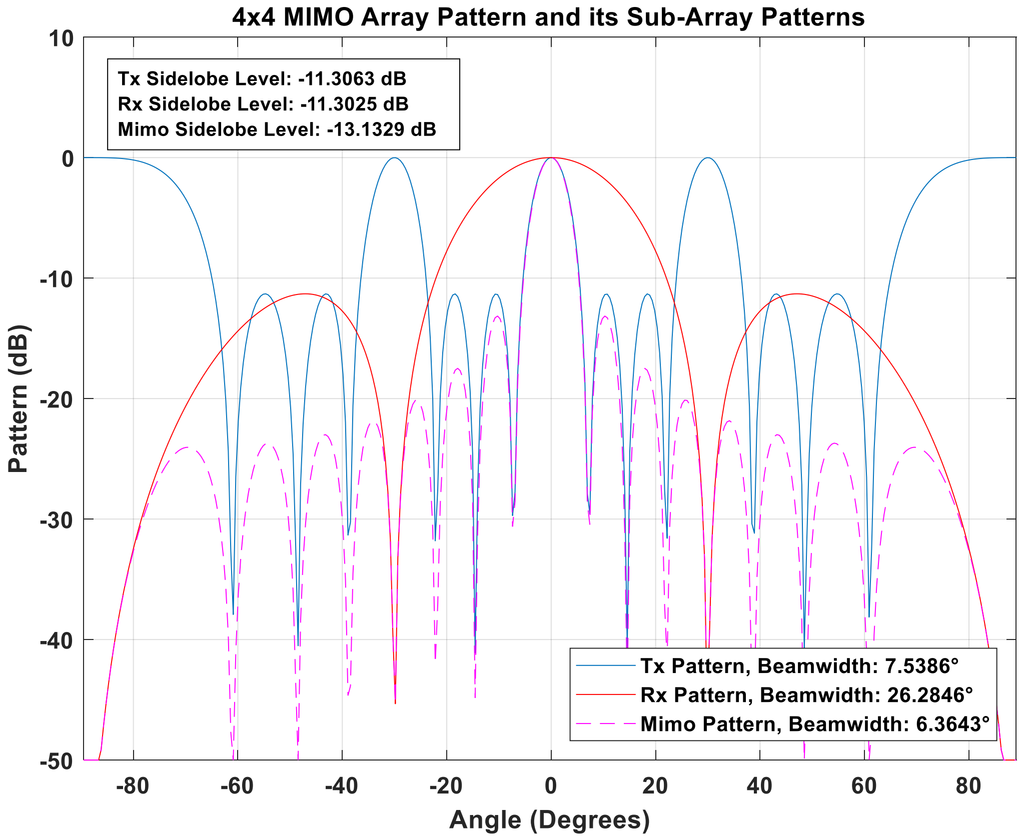


Figure 2-4 4x4 MIMO Array pattern designed by equation 2-13

With the virtual array concept, the aperture size now refers to the MIMO aperture, which corresponds to the virtual array aperture. When aperture size is mentioned, unless referred to as the physical aperture size, MIMO aperture often refers to the virtual array aperture size.

Equation 2-13 also means that each array can be steered to an angle on its own using the physical element positions. And although this approach slightly increases the computation load it leaves us with a more modular signal processing system.

$$AF_{mimo}(\theta_{steer}) = \sum_{n=1}^{N} w_{tx}(\theta_{steer})e^{-jkr_n} \sum_{m=1}^{M} w_{rx}(\theta_{steer})e^{-jkr_m} \qquad 2\text{-}14$$

For all sorts of beamforming purposes (steering, focusing, anti-jamming etc.) necessary complex weights can be calculated and multiplied separately in the data path for each of these virtual elements (Figure 2-5). This allows a MIMO array to be steered separately with transmit and receive beamforming to an arbitrary angle using the same steering methods to steer a uniform linear phased array as shown in 2-14;

$$w_{tx}(\theta) = e^{-j\frac{2\pi}{\lambda}\left((m-1)-\frac{M-1}{2}\right)\sin(\theta)dT} \quad \textit{2-15}$$

$$w_{rx}(\theta) = e^{-j\frac{2\pi}{\lambda}\left((n-1)-\frac{N-1}{2}\right)sin(\theta)dR} \quad \textit{2-16}$$

The steering weights are the same for steering transmit and receive arrays as if they're uniform linear phased arrays, which is shown again in 2-15 and 2-16 [53, p. 328]. Note that steering vectors in 2-15 and 2-16 are written assuming the angle θ is from the centre of the array to a target in far-field.

The summary of this approach can be observed in Figure 2-5, where it starts with where Figure 2-2 finishes. After extracting the virtual array element signals, they can be now used for beamforming as if they were received from an equivalent phased array. For beamforming, virtual array element signals would have to be steered with first receive array steering weights, and then the transmit array steering weights. However this is equivalent to computing a generic MIMO steering vector, where multiplication of the virtual element signal with this vector produces a phase shift equivalent to steering the virtual array directly to a specific angle. Therefore, steering vector necessary for MIMO beamforming can be calculated as the multiplication of steering vectors for transmit array and receive array. After the steering vectors are computed, they are then multiplied with the signals and all signals are summed to produce a single beam looking at a specific angle. The summation will be maximised if there's a target at the chosen angle because the phase shifts applied will make the summation of the signals add constructively. Repeating this process for all angles of interests then yields a range-angle map showing the reflection intensity at different range-angle pairs.

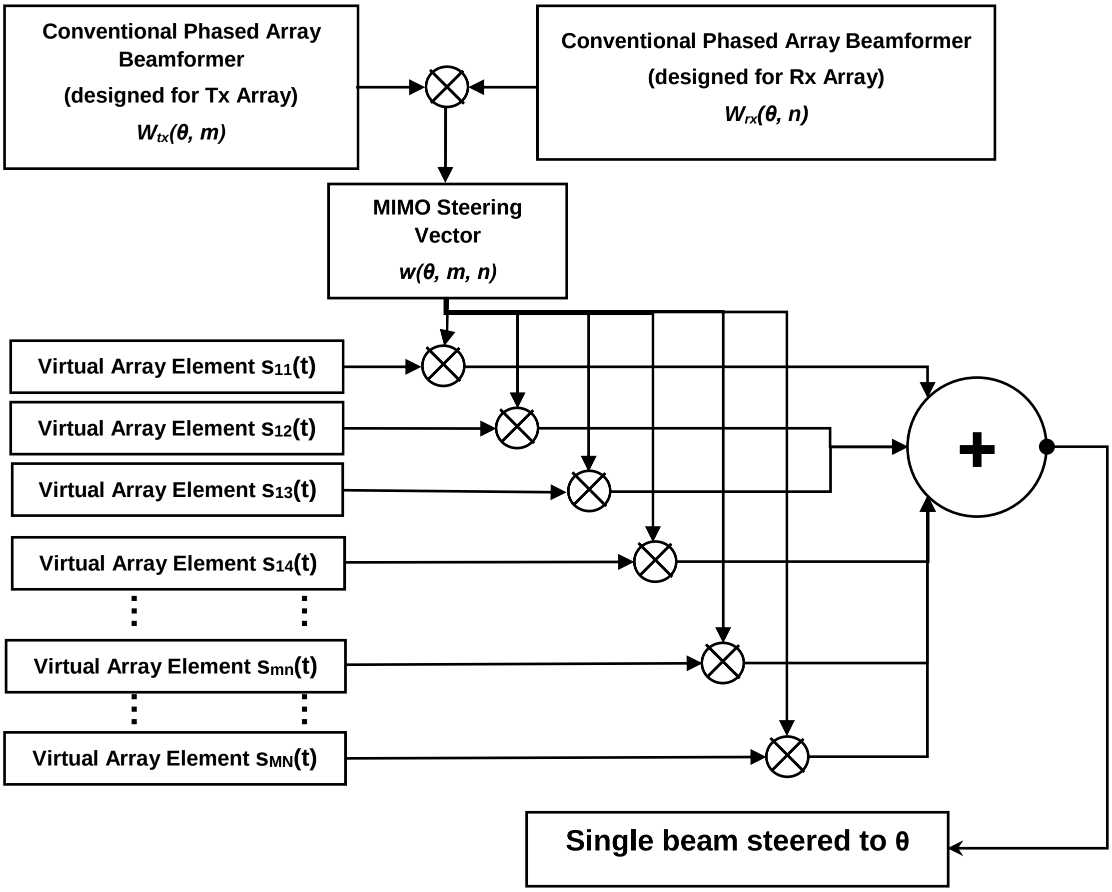


Figure 2-5 Block diagram for MIMO Beamformer at far field

2.3 SIMULATIONS

A MIMO radar simulator was built in MATLAB to confirm understanding of MIMO beamforming and as a mean of verifying experimental results obtained. Its major properties are adjustable input waveform, weighting window type selection in range, quantization level setting, sample rate setting, direct path transmission simulation, adjustable 3-dimensional antenna and target positions, and adjustable target RCS (radar cross section) levels. The main reason to implement such simulator was to be able to imitate experimental parameters as close as possible with the ability to simulate targets at chosen ranges. And this is achieved by simple ray tracing (without multipath). This is in comparison to simply using the derived array factor and computing the expected beam pattern (equation 2-14). This allows simulator to compute the near-field effects, direct-path signal (also referred to as direct signal in this thesis), sampling rate and even quantization without us having to derive them first. But if the conditions are ideal, then simulator's results should approximate to the derived model, and hence it would make a good validator for both the derived model and the experimental results.

The simulator works by simulating point targets with adjustable RCSs. A carrier-level transmit waveform is selected -which is used as a time-of-flight pulse- and the transmit wave's path is calculated from a transmitter to target and back to a receiver, all of which are defined with 3-dimensional Cartesian coordinates. The reason behind using carrier-level transmit waveforms is that these are the exact same vectors that are fed into the signal generators in the experimental setups.. These paths are then used to apply time delays, phase shifts and propagation losses to transmit waveform for each transmitter, receiver and target trio. However, the simulator does not compute possible multipaths between targets. The transmit waveform is time delayed, phase shifted and attenuated between transmitter and receivers to simulate direct path signals too, or sometimes referred to as the direct signals. Simulator software does all of these at a carrier level using a high sampling rate to imitate analog signals but then downsamples the signals to experimental system's sampling rate to imitate sampling occurring in experimental hardware. Additionally, digital filters and downconversion has been used to imitate other experimental hardware that is used. The software was written completely from scratch as at the time MATLAB's radar tools has been found to be insufficient for MIMO radar modelling. The simulator software codes can be found in .

For our simulations the following parameters in Table 2-1 were used since they are the closest parameters that was used to conduct our initial MIMO experiments. Two simulations have been run placing a target at an arbitrary position at far-field, with and without direct path transmission to observe the effects. A Gaussian window was used to reduce range sidelobes, which makes the range sidelobes outside of the dynamic range of the range-angle map. No weighting window was chosen to be used in azimuth because sidelobe levels indicate correct beamforming, as well as the beamwidth. Sample rate was chosen as 3.125 Gigasamples/s to match the sampling rate of the experimental equipment. Note that this sampling rate is not the rate that is used to simulate carrier-level signals. For carrier-level signal simulations, a sample rate of 10 Gigsamples/s has been used, because this is the sample rate of the experimental waveform generator. Finally, for the target RCS 50 m^2 is chosen because this value has been found to be a good example for showing direct path transmission effects in relation.

Table 2-1 Simulation parameters

Property	Value	Unit
Number of Tx	3	-
Number of Rx	5	-
Carrier Frequency	3.5	GHz
Wavelength	8.57	cm
Waveform	Upchirp LFM	-
Multiple Access Scheme	TDMA	-
Bandwidth	1	GHz
Range Window	Gaussian	-
Angular Window	None	-
Antenna Types	Isotropic	-
Tx Element Spacing	20	cm
Rx Element Spacing	4	cm
Fraunhofer Range	>2.5	m
Baseband Sample Rate	3.125	Gigasamples/s
Carrier-level Sample Rate	3.125	Gigasamples/s
Pulse Length	2	microseconds
PRI	1	milliseconds
Number of Pulses	500	-
Target RCS	50	m^2

Below is the MATLAB generated scenario view from top in Figure 2-6. The targets and antennas are all placed in the same XY plane. Even though the dimensions of our anechoic chamber is larger than 10 meters, in this simulation well beyond Fraunhofer distance, the target has been placed to ensure no near-field effect interferes with this simulation where we expect to see a near-ideal beam pattern. The target is placed at 10.7 m slant range with an azimuth angle of 13.4 degrees.

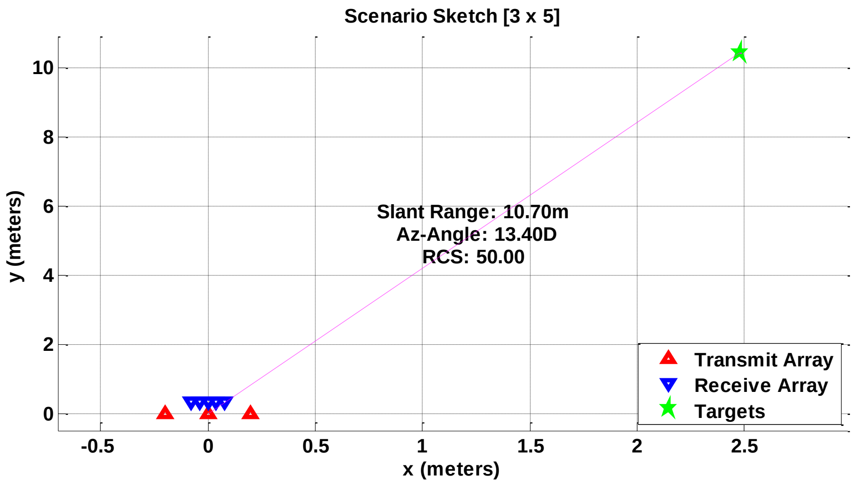


Figure 2-6 13.5 degrees target simulation scenario sketch

In the Figure 2-7 below the range-azimuth map obtained via running the simulations without direct path transmission can be observed. As would be expected the map shows only the target signature at just above 10 meters slant range and at 13.5 degrees azimuth. The azimuth cut of the map at 10.55 meters can be seen in Figure 2-8.

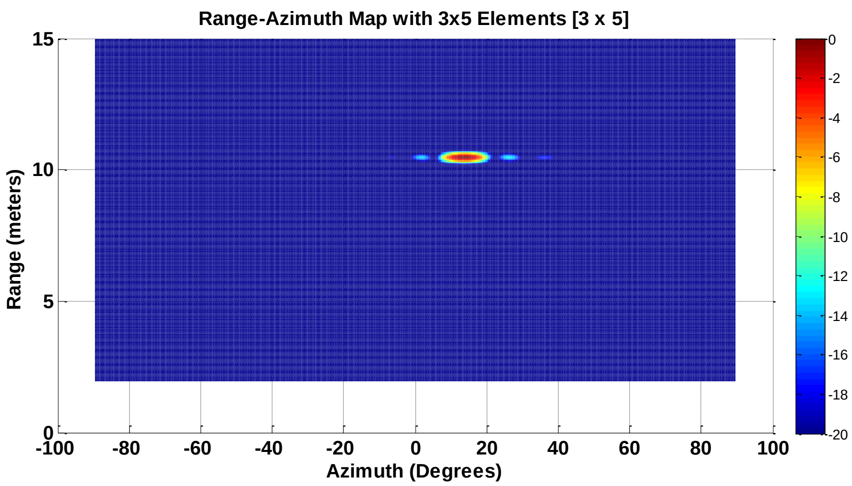


Figure 2-7 Range-Azimuth map of mimo radar simulation of a target at 13.5 degrees

In Figure 2-8 the azimuth cut at 10.5 meters from the range-angle map in Figure 2-7 can be seen. It can be observed that the beamwidth is measured as 7.48 degrees and the sidelobe level is at -12.9 dB.

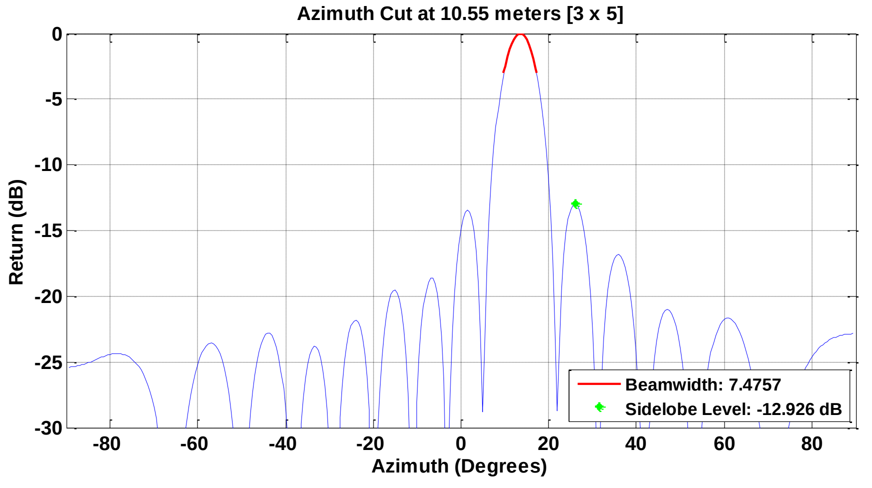


Figure 2-8 Azimuth cut of the above range-azimuth map at 10.55 meters

For the verification of our simulation programme, the expected MIMO array factor has been computed and plotted at the given range and angle using a MIMO array with these parameters using equation 2-14. In Figure 2-9, the expected beam pattern steered to 13.4 degrees at 10.5 meters can be seen. It can be observed that the beamwidth is measured as 7.44 degrees and the sidelobe level is at -12.9 dB. The expected beamwidth in radians can be approximated as $BW_{3dB} \cong 50\frac{\lambda}{L*\cos(\theta_{steer})}$ in degrees, where λ is the wavelength, L is the aperture length (in MIMO case this would be the virtual aperture length), and θ_{steer} is the scan direction [54, p. 58]. Using this approximation the expected beamwidth is found to be 7.34 degrees. These results coincide with the results we obtain from simulations, which also coincide with the theory. Therefore we concluded that our simulation results works nominally.

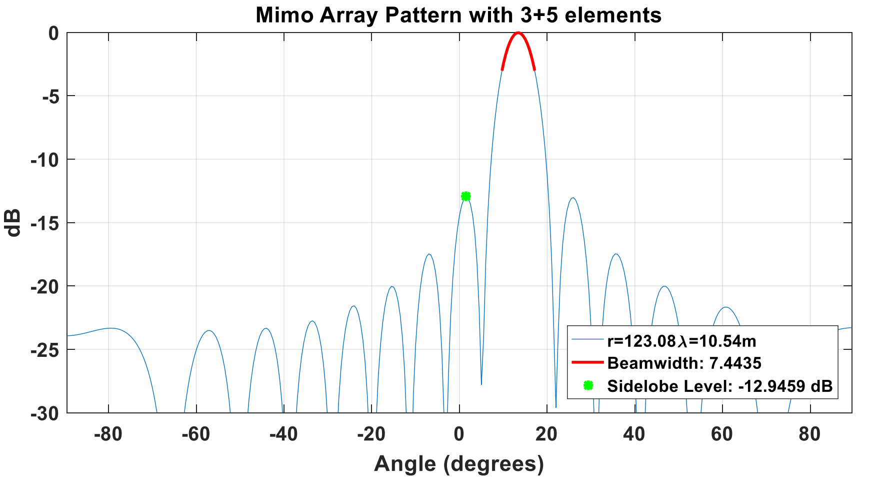


Figure 2-9 Computed (expected) MIMO array pattern with 3 transmit, 5 receive elements

However, results above are the simulation results without direct path (transmit to receive) transmissions simulated. From initial equipment tests, it was expected to have interference from direct path transmissions and therefore it was decided to include it in simulations. The simulation scenario above was run again, but this time with direct path transmissions enabled. Below are the results with the effects of direct path transmissions where we can observe the effect –especially- in close ranges.

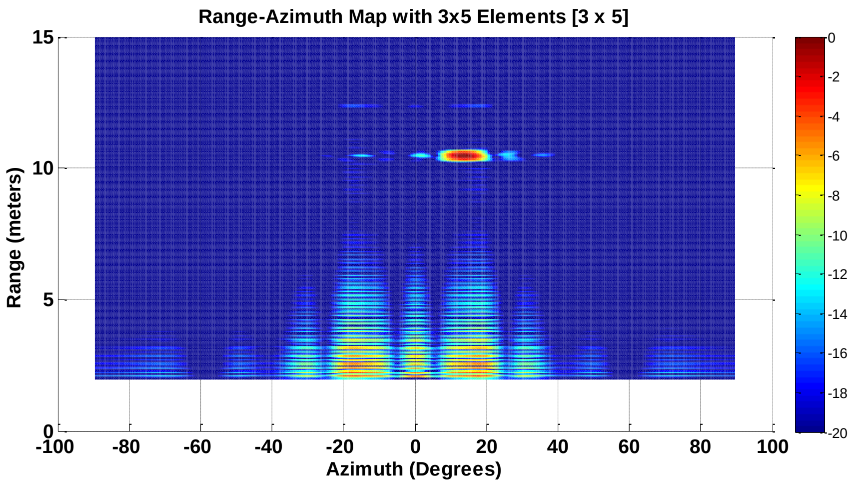


Figure 2-10 Range-Azimuth map of MIMO radar simulation of a target at 13.5 degrees (direct path transmissions switched on in simulation)

In Figure 2-10 above the range-azimuth map obtained via running the simulations with direct path transmission can be observed. The map now shows artefacts due to direct path transmission in addition to the target signature at just above 10 meters slant range and at 13.5 degrees azimuth. The azimuth cut of the map at 10.55 meters can be seen in Figure 2-11.

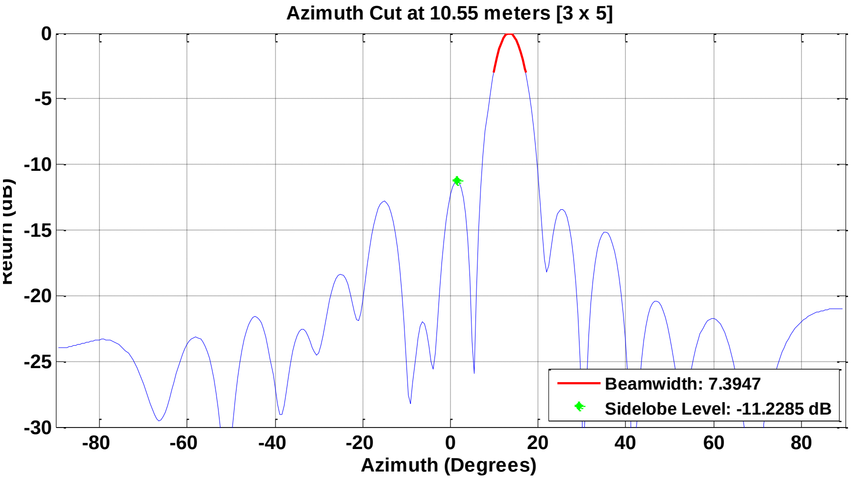


Figure 2-11 Azimuth cut of the above range-azimuth map at 10.55 meters

In Figure 2-11 the azimuth cut at 10.5 meters from the range-angle map in Figure 2-10 can be seen. It can be observed that the beamwidth is measured as 7.4 degrees and the sidelobe level is at -11.2 dB. Compared to the simulation without direct path transmissions, the beamwidth is almost preserved but the sidelobe levels decreased. The drop in sidelobe levels is due to the range sidelobes of the direct path transmission. This effect can be more clearly observed from the range-angle map following the range sidelobes in Figure 2-10. In this simulation, we observe that near-field range sidelobes of direct path transmissions can almost reach 10 meters and can almost interfere with the target signature even with a highly reflective target with 50 RCS at 10 meters range.

2.4 Experimental Results

2.4.1 Experimental System

Block Diagram

In our experimental system, a TDMA (time division multiple access) MIMO setup has been implemented via use of LFM pulses. This was used throughout the PhD study, to circumvent limitations of experimental equipment, including simultaneous signal transmission over multiple channels, relatively low receiver dynamic range, and sample rate requirements. In addition, TDMA is theoretically the best option compared to FDMA (frequency division multiple access), CDMA (code division multiple access) or any other multiple access schemes because it can provide real orthogonality among transmit signals. This allowed for a better assessment of beamforming performance throughout this research. It also allows re-use of the same waveform, which means we can make full use of the available bandwidth, and wouldn't have to worry about summing possibly different matched filter outputs.

We have performed the first set of experiments with our existing equipment in the far-field region to test our theory and algorithms and to verify our simulations. To generate compressed waveforms a Tektronix AWG7102 arbitrary waveform generator was used[55]. And to capture the reflected signals, a Tektronix DPO72004C digital phosphorus oscilloscope was used[56]. The captured signals were then transferred to MATLAB in a desktop computer for processing. Due to sampling issues that DPO (Digital {Phosphorus Oscilloscope) was having at the time of experiments, signals were downconverted to a low-IF band of 500 MHz and then recorded with DPO. The reason behind this was to put the signal band (1GHz bandwidth) into a real region where we could still sample within Nyquist criteria. So a 1GHz bandwidth signal with a centre frequency of 0.5GHz was directly sampled (no IQ conversion, single channel real sampling) with a rate 3.125 Gigasamples/s. For the purposes of downconverting a coherent signal generator was used[57]. The recorded low-IF signals were then downconverted to baseband digitally in MATLAB. Transmission of the signals were done via a directional horn antenna. Receive antennas were custom design omnidirectional antennas built by other colleagues in our research laboratory. A system block diagram summarising the experimental setup can be inspected in Figure 2-12.

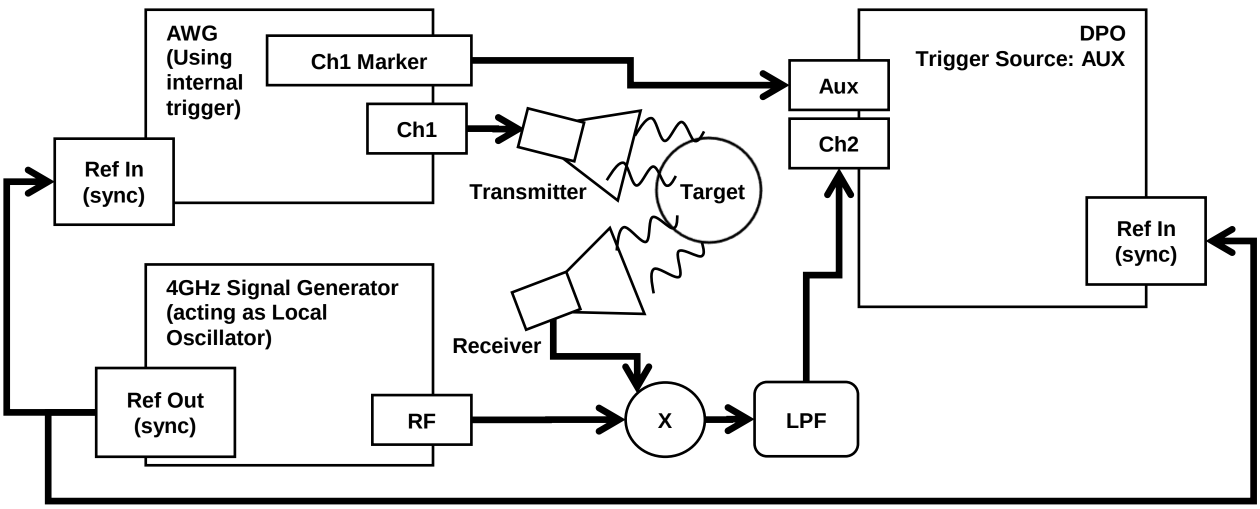
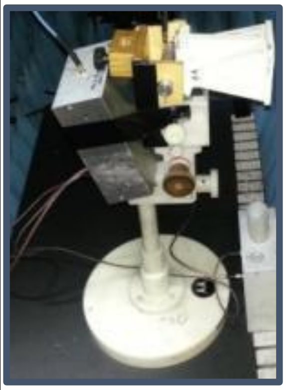
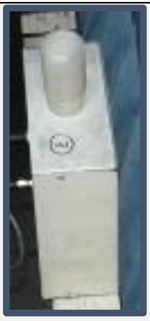
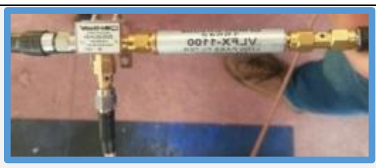


Figure 2-12 System block diagram of initial radio frequency MIMO experimental setup

The final setup was reached through various trials beforehand. These trials were made to see if and how the existing laboratory equipment could be used for MIMO experiments. The trials started with single transmitter and a single receiver simple radar configuration, and the setup was refined until it became feasible. Later on, number of transmitters and receivers were increased step-by-step to reach a feasible TDMA MIMO setup. The details of these trials will not be discussed here but can be found in Appendix A under the title "Initial MIMO Experimental System Development". The photos and relevant details of the equipment can be found in Table 2-2. And the final experiment parameters can be seen in Table 2-3.

Table 2-2 Experimental setup's bill of materials

Equipment	Photo	Notes
Tektronix AWG7102 Arbitrary waveform generator		• 10 GS/s sample rate • Internal trigger • External reference (uses the signal generator's reference 10 MHz) • Sends a marker pulse
Transmitter module		• 11dB power amplifier • Transmit antenna • 120 degrees 3dB-beamwidth (from specs)

Receiver module		• Omnidirectional receive antenna • Receive amplifier box • 20 dB gain • 3-4.3 GHz bandpass filter
HP 8648D Signal generator		• Locked to 2.97 GHz • Always running • Using internal reference
RF downconverter module		• RF Downconverter • 1 GHz low-pass filter
Tektronix DPO72004 Digital Phosphorus Oscilloscope		• 3.125 GS/s per channel • External trigger (uses the marker from AWG) • External reference (uses the signal generator's 10 MHz reference)
Calibrated target		• 22 cm radius metal ball

As previously mentioned the parameters below in Table 2-3 were reached through a number of trials. The number of transmitters and receivers were chosen in order to minimise the beamwidth (aperture size). The carrier frequency was chosen as 3.5 GHz because, our research laboratory already had amplifier and antenna equipment around that centre frequency. These include wideband and widebeam horn antennas for transmission and in-house custom-built omnidirectional antennas. Both of the antenna specs were previously verified by the members of our laboratory. Also included in the inventory were off-the-shelf amplifier and bandpass circuits from Mini-Circuits, for gain and attenuation properties their datasheets were referred to. The bandwidth was chosen as high as possible, but was limited by the bandpass limiters we had, which was 1 GHz. An LFM signal was used to increase SNR simply by increasing pulse-length if needed. A Gaussian range window was used to avoid range-sidelobes without degrading the range-resolution too much. Other parameters listed below were chosen either out of necessity (such as transmit antenna beamwidth) or found via trials (such as the number of pulses to maximise SNR).

Table 2-3 Experimental setup parameters

Property	Value	Unit
Number of Tx	3	-
Number of Rx	5	-
Carrier Frequency	3.5	GHz
Waveform	Rectangular Upchirp LFM	-
Bandwidth	1	GHz
Low-IF	500	MHz
Range Window	Gaussian	-
Angular Window	None	-
Tx Antenna Type	Horn	-
Tx Antenna Gain	8	dBi
Tx Antenna Beamwidth	60	Degrees
Tx Power Out	25	dBm
Rx Antenna Type	Omnidirectional	-
Rx Antenna Gain	2	dBi
Rx Antenna Beamwidth	N/a	-
Receive Gain	24.5	dB

Tx Element Spacing	20	cm
Rx Element Spacing	4	cm
Fraunhofer Range	>2.5	m
Sample Rate	3.125	Gigasamples/s
Number of Pulses	500	-

2.4.2 Scenarios and Results

Various experiments were performed where target(s) were placed about 3.2m away from the radar and changing azimuth angles. Below are the 3 scenarios simulated with various configurations and experimented with a 3x5 MIMO radar. Next to the top view sketches of setups are the photos taken inside the anechoic chamber during experiments. These scenarios are designed to check beamforming and target localization capabilities, azimuth resolution and finally multiple target resolution capabilities.

Also below are the range-angle maps obtained from experiments along with azimuth cuts at primary target ranges. Along with them are corresponding simulation results; range-angle maps with azimuth cuts at primary target ranges obtained from simulating the experimental scenarios. After the discussion of results, comparisons are made between experiment and simulation results.

Single Target

In Figure 2-13 the over the top sketch of the first scenario can be seen. A target is placed at about 3.5m slant range with about -8 degrees angle. The photo of the setup can be seen in Figure 2-14. This scenario was designed to verify single target detection capabilities (beamwidth, sidelobe levels, etc.), and to test left-right ambiguities.

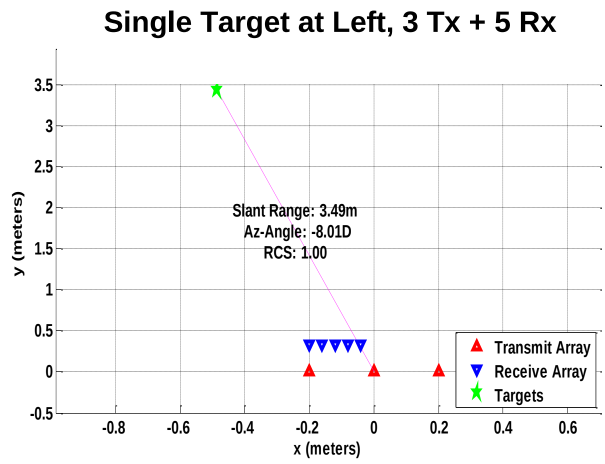


Figure 2-13 Single Target at left, top view of scenario

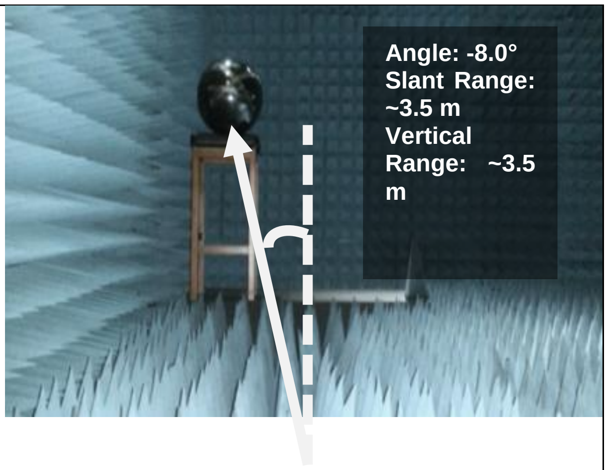


Figure 2-14 Single target at left, photo of the experiment

In Figure 2-15 is the range-angle map obtained from the 1st experiment where a single target was placed slightly to the left of the boresight. A single return can be located around 3.4 meters and at about -6 degrees. In Figure 2-16 an azimuth cut of Figure 2-15 at 3.4 meters can be seen. The measured beamwidth is about 7.3 degrees and the target's nearest sidelobe is at about -11.4 dB.

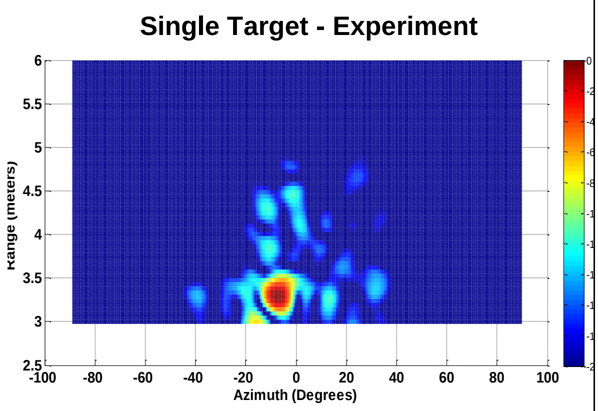


Figure 2-15 Range-angle map from experiment with single target

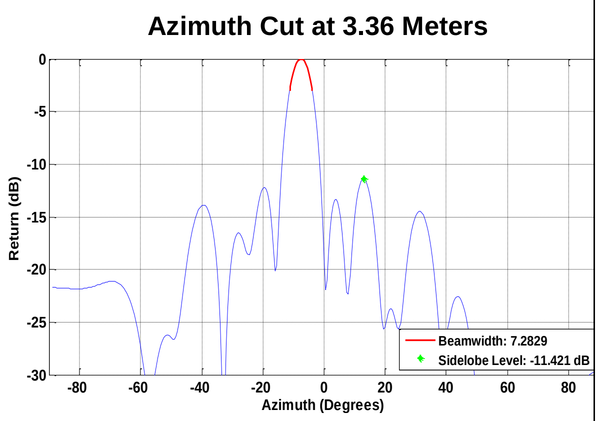


Figure 2-16 Azimuth cut at 3.3 meters from experiment with single target

In Figure 2-17 is the range-angle map obtained from simulating the scenario of the 1st experiment. A single return can be located around 3.4 meters and at about -6 degrees. In Figure 2-18 an azimuth cut of Figure 2-17 at 3.4 meters can be seen. The measured beamwidth is about 7.2 degrees and the target's nearest sidelobe is at about -9.3 dB.

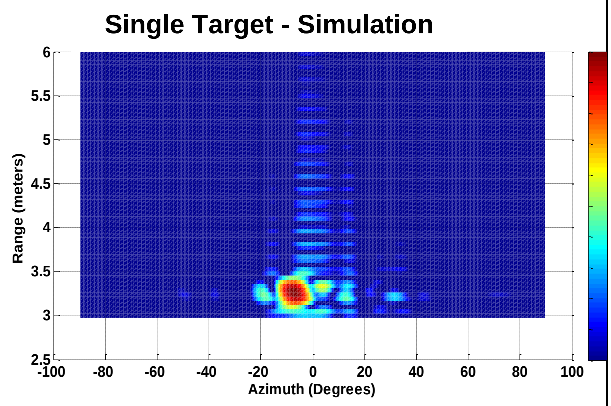


Figure 2-17 Range-angle map from simulation with single target

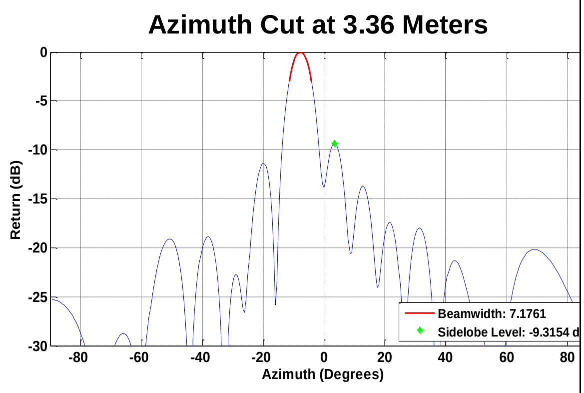


Figure 2-18 Azimuth cut at 3.3 meters from simulation with single target

When experiment and simulation results are compared, in both results the target can be seen at the correct slant range and azimuth angle. The measured beamwidths match to what is expected from this MIMO configuration too (see Figure 2-9 for a reference). The experimentally measured sidelobe level is within expected range of -13.1 dB level, however the simulation measured sidelobe level is lower than both. This is due to the way direct path transmission is simulated within the simulation programme. The simulation programme does not take into account the absorption losses or other background reflections from inside the anechoic chamber. Therefore it is expected to have slightly different levels. However what's also important to notice is the location (angles) of the sidelobes. It can be seen that even though the levels are not the same, the locations of the sidelobes match, which tells us that the beamforming algorithms work as expected.

Two Targets at Same Range

In Figure 2-19 the over the top sketch of the second scenario can be seen. Two targets were placed at about 3.5m slant range with about -6.5 and +6 degrees angles. The photo of the setup can be seen in Figure 2-20. This scenario was designed to verify multiple target resolution capabilities at same range and to observe how they interact.

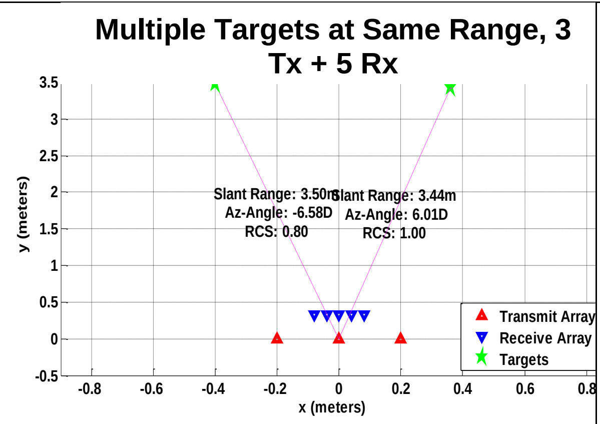


Figure 2-19 Multiple targets at same range, top view of scenario

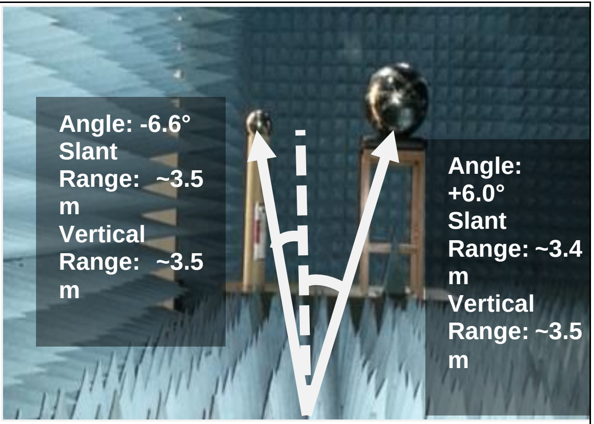


Figure 2-20 Multiple targets at same range, photo of the experiment

In Figure 2-21 is the range-angle map obtained from the 2nd experiment where two targets were placed at the same slant range with different azimuth angles. Two returns can be located around 3.4 meters and at about -6.5 degrees and 6 degrees. In Figure 2-22 an azimuth cut of Figure 2-21 at 3.4 meters can be seen. The measured beamwidth is about 7.3 degrees. From the azimuth cut, the strength difference between the targets can be measured as -4.5 dB. There was no attempt to measure the sidelobe levels due to target signature superposition in the azimuth cut (i.e. main lobe of second target is overlapping with first target's sidelobes and vice versa).

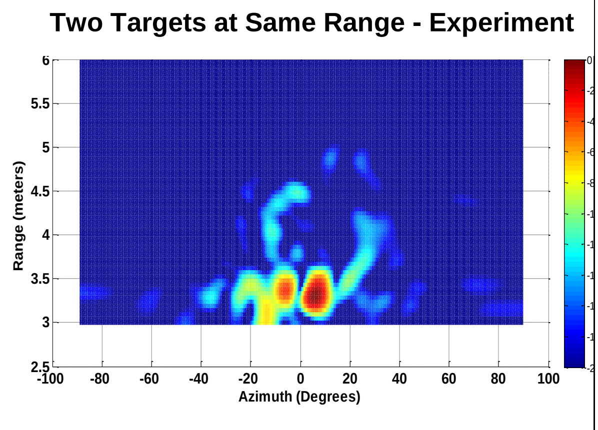


Figure 2-21 Range-angle map from experiment with two targets at same range

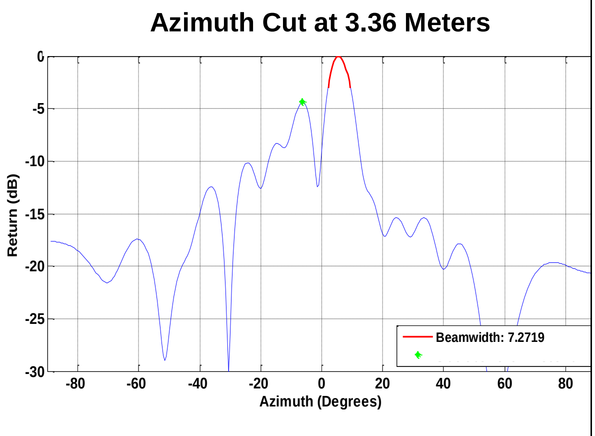


Figure 2-22 Azimuth cut at 3.3 meters from experiment with two targets at same range

In Figure 2-23 is the range-angle map obtained from simulating the scenario of the 2nd experiment. Two returns can be located around 3.4 meters and at about -6.5 degrees and 6 degrees. In Figure 2-24 an azimuth cut of Figure 2-23 at 3.4 meters can be seen. The measured beamwidth is about 7.3 degrees. From the azimuth cut, the strength difference between the targets can be measured as about -4.5 dB.

Again, there was no attempt to measure the sidelobe levels due to target signature superposition in the azimuth cut (i.e. main lobe of second target is overlapping with first target's sidelobes and vice versa).

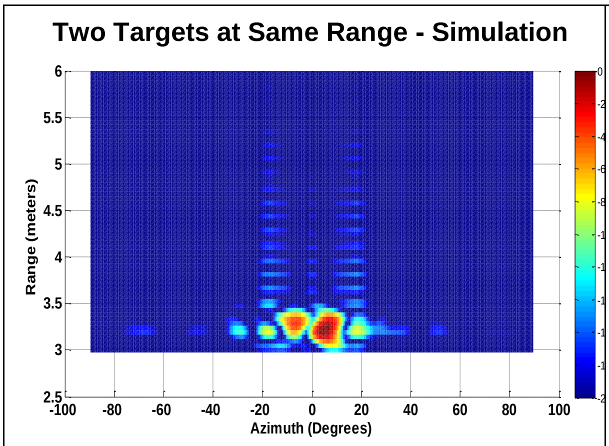


Figure 2-23 Range-angle map from simulation with two targets at same range

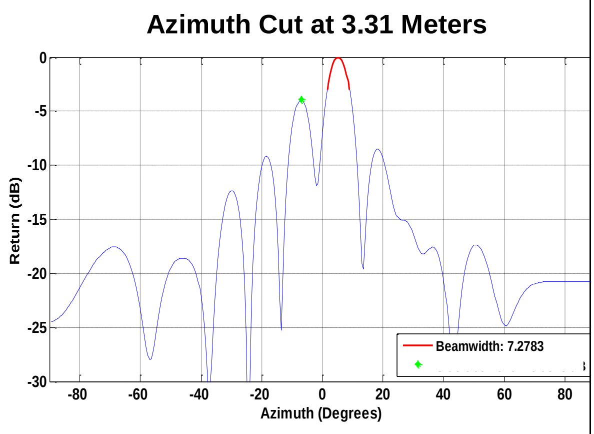


Figure 2-24 Azimuth cut at 3.3 meters from simulation with two targets at same range

When experiment and simulation results are compared, in both results both targets can be seen at the correct slant range and azimuth angles. The measured beamwidths match to what is expected from this MIMO configuration too (see Figure 2-9 for a reference). It's important to note that even though we have two targets at the same range occupying the same range bin, we can resolve them in angle, this becomes especially clear when compared with a return from a single target (see Figure 2-17 for reference).

Two Targets at Different Range

In Figure 2-25 the over the top sketch of the third and last scenario can be seen. Two targets were placed at about 2 meters and 3.5 meters slant range with about -6 and +4 degrees angles respectively. The photo of the setup can be seen in Figure 2-26. This scenario was designed to verify multiple target resolution capabilities at different ranges and to observe how they interact.

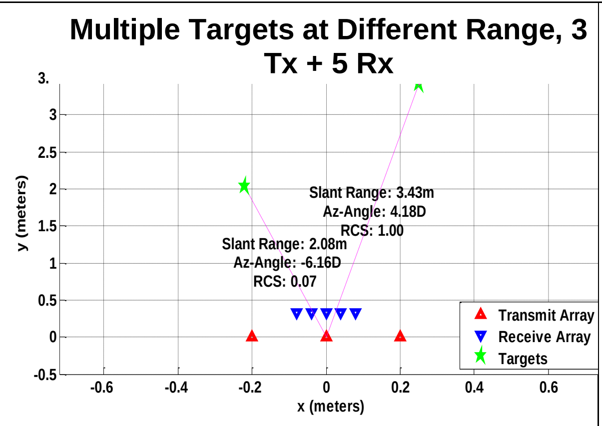


Figure 2-25 Multiple targets at different range, top view of scenario

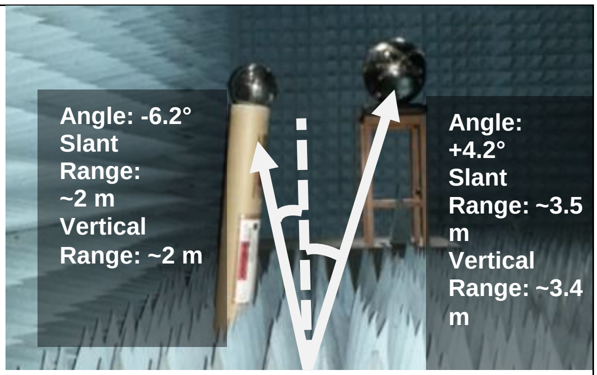


Figure 2-26 Multiple targets at different range, photo of the experiment

In Figure 2-27 is the range-angle map obtained from the 3rd experiment where two targets were placed at the same slant range with different azimuth angles. Two returns can be located around 2 meters and 3.4 meters, at about -6 degrees and 4 degrees. From the range-angle map the strength difference between the targets can be measured as -5 dB. In Figure 2-28 an azimuth cut of Figure 2-27 at 3.4 meters can be seen. The measured beamwidth is about 7.5 degrees and the nearest sidelobe level is measured to be at -7.5 dB. The increased sidelobe level can be easily explained by the existence of another target at a different range and that its range sidelobes overlap with first target's angle sidelobes.

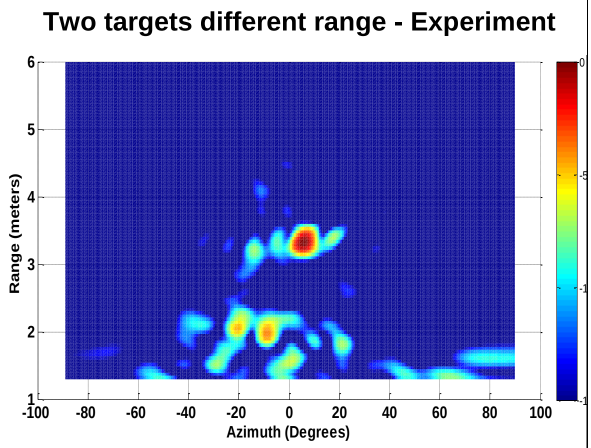


Figure 2-27 Range-angle map from experiment with two targets at different ranges

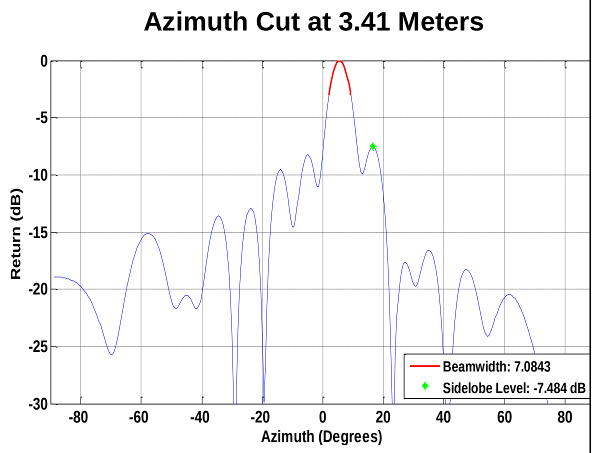


Figure 2-28 Azimuth cut at 3.4 meters from experiment with two targets at different ranges

In Figure 2-29 is the range-angle map obtained from simulating the scenario of the 3rd experiment. Two returns can be located around 2 meters and 3.3 meters and at about -4 degrees and 6 degrees. One important feature to note is the shape of the nearer return in the range-angle map. The target signature is effected by the direct path transmission and possibly by near-field effects too (Fraunhofer distance for the coarse array was calculated to be 2.5 meters, see Table 2-3). Also from the range-angle map the strength difference between the targets can be measured as about -4.5 dB. In Figure 2-30 an azimuth cut of Figure 2-29 at 3.3 meters can be seen. The measured beamwidth is about 7.7 degrees and the nearest sidelobe level is measured to be at -10 dB. Again, the increased sidelobe level can be explained by the existence of another target at a different range and that its range sidelobes overlap with first target's angle sidelobes.

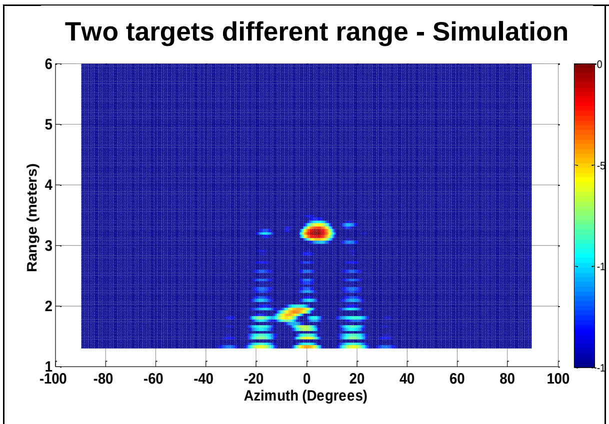


Figure 2-29 Range-angle map from simulation with two targets at different ranges

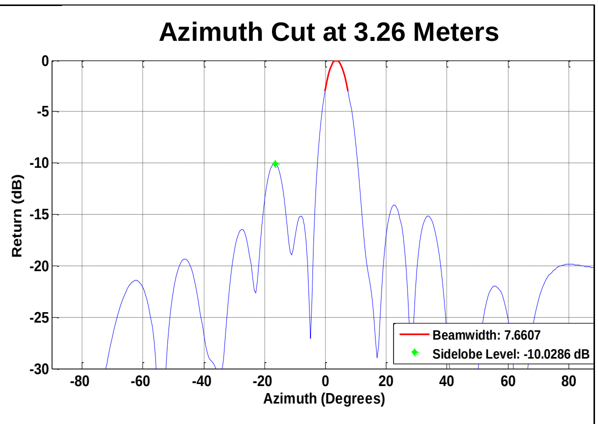


Figure 2-30 Azimuth cut at 3.3 meters from simulation with two targets at different ranges

When experiment and simulation results are compared, in both results both targets can be seen at the correct slant range and azimuth angles. The measured beamwidths still match to what is expected from this MIMO configuration (see Figure 2-9 for a reference). One key aspect to note that even though we have two targets at different ranges and different angles, and that we can resolve them separately in range and in angle, their sidelobes still interact with each other causing increases in angular sidelobes (possibly in range sidelobes as well, but this phenomenon is not discussed here). This becomes especially clear when compared with a return from a single target (see Figure 2-17 for reference). Another thing to note is the effect of direct path transmissions in nearer objects; the shape of the reflection from the second (nearer) target has a parallelogram shape. This happens due to the relatively strong range sidelobes of the direct path transmission interfering with the target signature, which makes it difficult to resolve smaller (in terms of RCS) objects in nearer ranges.

2.5 CONCLUSIONS

The fundamental theory of MIMO radar has been studied. This includes the derivation of MIMO array factor and beam steering using the base knowledge of phased array factor and phased array digital beam steering.

MIMO array factor computations and beamforming algorithms have been coded in MATLAB. And a simulation programme has been coded in MATLAB to test the algorithms and to accompany the experimental results for comparison. The simulation programme is first verified via using the mathematical derivations.

An experimental system has been built to test the MIMO beamforming theory. The experimental setup was based on readily available RF equipment as a first confirmation prior to the development of a purpose specific technology demonstrator. Experiments were done in an anechoic chamber to verify real-life beamforming capabilities using our MIMO beamforming algorithms. These experiments included scenarios containing one or two targets at different ranges to prove various target resolving capabilities.

All experimental scenarios were simulated using the previously confirmed simulation programme for comparison of experimental results. It was found that the MIMO array beamforming works well within expected parameters in a controlled environment, and that the experimental results match simulation results within nominal deviations. The obtained results confirmed understanding of MIMO radar theory and the feasibility of our approach for the next steps which is the application of MIMO array radar in near-field.

3 MIMO SENSOR ARRAY IN NEAR-FIELD

3.1 INTRODUCTION

Like phased array systems, MIMO arrays also have their limitations as targets get nearer. Usually, array sensing systems are designed in a way so that the target area of interest lies in the far-field of the array. For automotive sensing purposes, this may not always be the case and short-range sensing in a vehicular environment implies sensing in the near-field of the array. In the previous chapter an analysis of MIMO array factor in far-field has been done and verified via experiments and a custom simulation programme. During the derivation of the MIMO array factor, far-field approximations were used; such as assuming –array– element to target angles remain the same and that the range lines running from array elements to target are parallel. However, as the targets gets closer, these approximations start to fail. Therefore, conventional beamforming and steering techniques depending on these approximations start to fail as well.

Theoretically, it may be plausible that a MIMO array can apply near-field corrections, as each virtual array element can be accessed individually and the appropriate phase corrections can be applied. However, such hypothesis should be theoretically derived and experimentally tested. Therefore, a similar approach to previous chapter will be taken in this chapter; element phase differences for an arbitrary MIMO array will be derived at near-field. Meaning that the approximations that were resorted before will not be used. Then, how MIMO near-field beam patterns can be digitally focused in near-field ranges will be shown by using the exact element phase differences (compared to a far-field approximated element phase differences). Finally, application of these focusing techniques will be presented via experiments and compared to corresponding simulations.

First section introduces the uniform linear array factor in near-field and derives the near-field MIMO array factor. The beam patterns for phased arrays and MIMO arrays were computed using the formulated array factors and presented at various ranges. Then, existing phased array near-field focusing methods are explained and a MIMO near-field focusing algorithm is developed based on the ideas explained. Then beam patterns for a phased array and a MIMO array were computed and compared with a near-field focused MIMO array beam pattern at an arbitrary near-field range. Later on, the accompanying simulation programme and experimental setup are explained, which were kept the same as previous chapter. Afterwards, experimental results with spherical targets in an anechoic chamber are presented with their corresponding simulations to confirm and show improvements. Finally, conclusions are presented to move on to next chapter where we discuss the design and development of a purpose specific MIMO technology demonstrator.

3.2 MIMO ARRAY FACTOR DERIVATION IN NEAR-FIELD

For the purposes of this section, the near-field phased array factor is derived rather than an entire MIMO array. The MIMO array factor is still the multiplication of the array factors of transmit and receive array. So, this derivation is then substituted in MIMO array factor, both for transmit sub-array and receive sub-array to yield near-field MIMO array factor. Similarly, any technique to change the beam pattern of transmit and receive sub-arrays can be used to alter (focus) MIMO beam pattern.

3.2.1 Geometry of Uniform Linear Array in Near-Field

The structure of a uniform linear array can be generalized for our purposes as below in Figure 1-2. The array elements and the target(s) are all assumed to be on the same plane (XY plane). Phase centre of the array is taken as the physical centre of the array to cover the cases of both odd and even numbered arrays. In Figure 1-2 below is a sketch of a near-field array configuration without the far-field approximations.

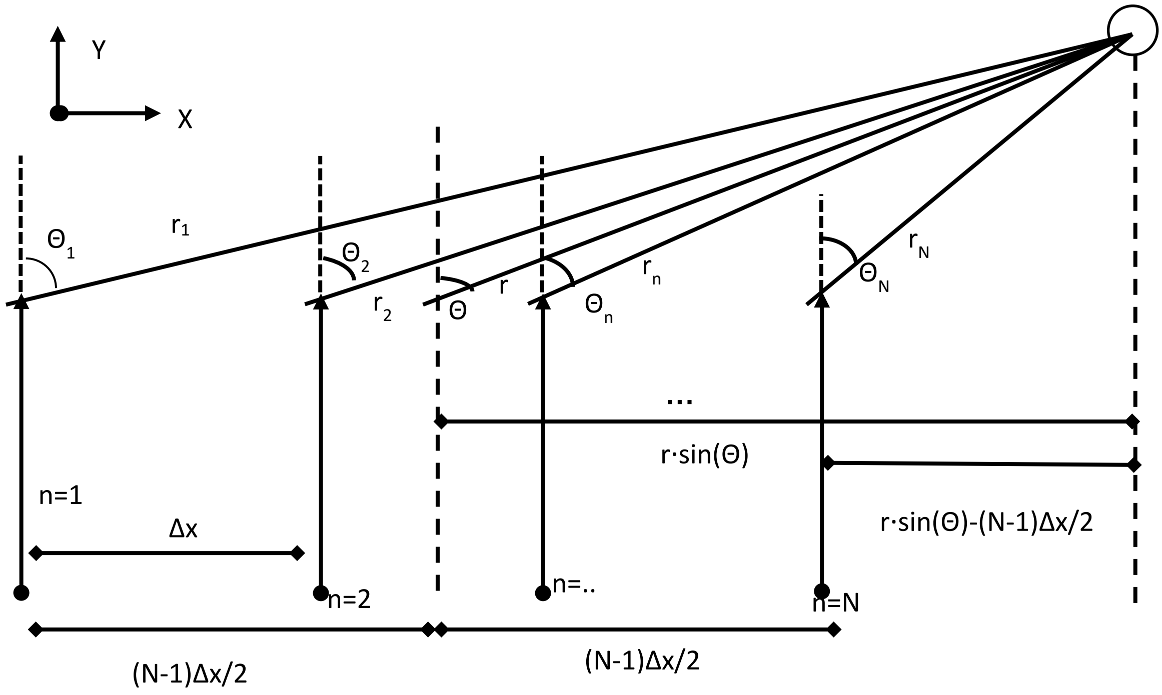


Figure 3-1 Generic uniform linear array geometry

To help with further calculations some variables have been written in terms of others. In equations that follow; n represents an arbitrary array element number where N represents total number of array elements (so n can take values between 1 and N inclusive). Δx is represents the uniform distance between each array element in electrical lengths (in terms of wavelength). Θ represents the azimuth angle from the centre of the array to an arbitrary target. Θ_1, Θ_2 and similarly Θ_n stand for specific azimuth angles from corresponding array elements to target. Similarly, \`$r$\` stands for the range from the centre of the array to an arbitrary target, whereas r_1, r_2 and r_n stand for specific ranges from corresponding array elements. Below are some of these geometric derivations which will be used further on. Most important of these are 3-3 & 3-7 which helps calculate the specific range & azimuth angle to a target using only the angle from centre to target (Θ), range to target from centre (r), element spacing (ΔX), element number (n), and total number of elements (N).

$$\tan(\theta_n) = \frac{r * sin(\theta) - \frac{(N-1)\Delta x}{2} + (N-1)\Delta x - (n-1)\Delta x}{r * \cos(\theta)} \qquad \text{3-1}$$

$$\tan(\theta_n) = \frac{r * sin(\theta) + \frac{(N-1)\Delta x}{2} - (n-1)\Delta x}{\mathrm{r} * \cos(\theta)} \qquad \text{3-2}$$

$$\theta_n = \operatorname{atan}\left(\frac{r * sin(\theta) + \frac{(N-1)\Delta x}{2} - (n-1)\Delta x}{r * \cos(\theta)}\right) \qquad \text{3-3}$$

Tangent of Θ_n can be written simply by looking at the sketch as 3-1. After simplification, it can be left as 3-2. Taking the inverse of the tangent yields Θ_n alone on the left side, ready for substitution as in 3-3.

$$\mathrm{r_n} * \cos(\theta_n) = \mathrm{r} * \cos(\theta) \qquad \text{3-4}$$

$$\mathrm{r_n} = \frac{\mathrm{r} * \cos(\theta)}{\cos(\theta_n)} \qquad \text{3-5}$$

Since all elements share the same range in Y dimension we can write 3-4 and then leave r_n alone on left hand side as in 3-5.

$$\mathrm{r_n} = \mathrm{r} * \cos(\theta) * \frac{1}{\cos\left(\operatorname{atan}\left(\frac{r * sin(\theta) + \frac{(N-1)\Delta x}{2} - (n-1)\Delta x}{r * \cos(\theta)}\right)\right)} \qquad \text{3-6}$$

$$r_n = \sqrt{r^2 + \sin(\theta)\,(N-2n+1)\Delta x + \frac{(N-2n+1)^2(\Delta x)^2}{4}} \qquad \text{3-7}$$

Then 3-3 can be used to substitute Θ_n into 3-5 as shown in 3-6. Intermediary steps are skipped since they are merely simplifications, and 3-7 is obtained.

3.2.2 **Uniform Linear Array Factor in Near-Field**

At near field, the assumption of having the angles from sensor elements to a target being approximately equal to each other doesn’t hold. So, the array factor remains as is and does not approximate into a Sinc function. Especially at nearer ranges the shape of the beam pattern starts to deviate from far-field patterns. This was only possible due to far-field assumptions where element ranges to a target could be written as in 1-7. And only then when 1-7 is substituted in 3-9 which is the array factor for a uniform linear array a Sinc function could have been obtained. But, if the exact range r_n is used as derived above in 3-7, the beam pattern can still be computed numerically even though it cannot be approximated to a known function.

$$r_{nfar} = r + \left(\frac{N-1}{2} - (n-1)\right)\Delta x sin(\theta) \qquad \textit{3-8}$$

$$AF(\theta) = \sum_{n=1}^{N} w_n(\theta) * e^{-jkr_n} \qquad \textit{3-9}$$

In Figure 3-2 below, numerically computed beam patterns of a 15-element phased array with half-wavelength element spacing reflected off of an ideal point target at various ranges using the array factor in 3-9 with the definition of r_n in 3-7 can be seen. The Fraunhofer distance (equation 1-6) for such an array is 98λ where λ (lambda) is the wavelength. At a reasonable far range such as 10000λ, it can be seen in the blue legend of Figure 3-2 that the beam pattern indeed looks like a Sinc function. At 98λ range (red legend), it can be seen that even though the shape of the beam is still very similar to a Sinc function, the nulls of the beam start to rise. This would immediately start to cause problems in MIMO beamforming, because in conventional MIMO beamforming nulls of the fine array cancels out the grating lobes of the coarse array. At 50λ range (orange legend) it can be observed that the nulls quickly start to disappear along with a slight increase in sidelobe levels. At 20λ range (magenta legend) it can be seen that beam first nulls are lost, and that there is even gain loss in the steering direction, the beam is now currently unusable. Finally at 10λ range (green legend), it can be observed that the beam pattern breaks. To put these ranges into context; wavelength that's been used in previous chapter's experiments was 8.6 cm. The Fraunhofer distance for this wavelength is 8.4 meters, and 50λ range where nulls are significantly lost is 4.3 meters. There is also a significant loss in gain that's observable, almost 8 dB below maximum at 10λ range.

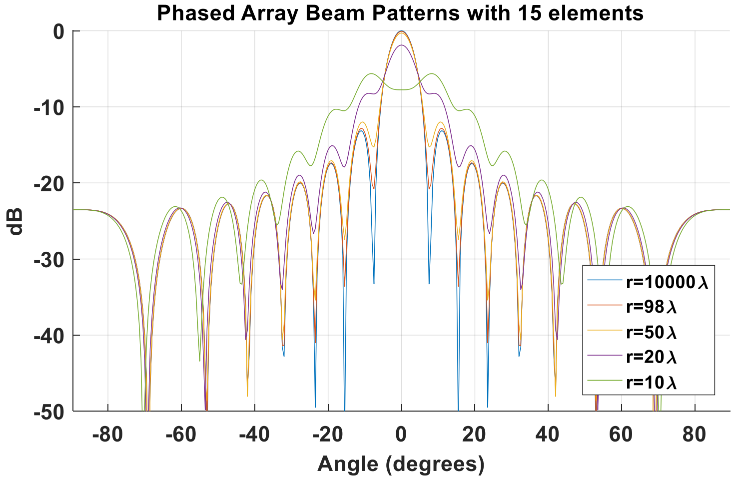


Figure 3-2 Uniform linear phased array factor in at various ranges (in terms of electrical lengths)

3.2.3 Uniform Linear Array Focusing Techniques

Even though beam patterns do not take the shape of a Sinc function at nearer ranges, the beam pattern can still be altered via means of digital beamforming techniques. These techniques -which are similar to beam steering in nature- are simply applying complex weights to received signals to shift the received phases to far-field phases.

$$f(\theta,n,r) = e^{jk(r_n - r_{nfar})} = e^{-jkr_{nfar}} * e^{jkr_n} \tag{3-10}$$

$$f(\theta,n,r) * e^{-jkr_n} = e^{-jkr_{nfar}} \tag{3-11}$$

A compensation method based on the phase differences of ideal far-field and an arbitrary range can be written simply as in 3-10 such that the addition of received phases with the phases of this filter would yield ideal far-field phases as shown in 3-11.

$$AF(\theta,r) = \sum_{n=1}^{N} w_n(\theta) * g(\theta,n,r) * e^{-jkr_n} \tag{3-12}$$

$$AF_{far}(\theta,r) = \sum_{n=1}^{N} w_n(\theta) * e^{-jkr_{nfar}} \tag{3-13}$$

In equation 3-12 and further on, g stands for the phase correction function and w_n stands for beam-steering weights. And this filter can be plugged into the array factor as the near-field focusing filter as in 3-12 to obtain 3-13 which is the ideal array factor at far-field.

These techniques are known and already used for near-field focusing (or sometimes referred to as lensing) for phased arrays. Examples and variations of these techniques can be found in literature. For example the method of having an extra set of complex weights (other than beam steering weights) has been proposed as below in 3-10 in order to obtain a desired far field beam pattern [17].

$$r_{nfar} = r + \left(\frac{N-1}{2} - (n-1)\right)\Delta x sin(\theta) \qquad \text{3-14}$$

$$r_n - r_{nfar} = \sqrt{r^2 + \Delta x \sin(\theta)\,(N-2n+1) + \frac{(N-2n+1)^2(\Delta x)^2}{4}} - r + \left(\frac{N-2n+1}{2}\right)\Delta x sin(\theta) \qquad \text{3-15}$$

Starting with the array factor for a phased array pattern and introducing near-field focusing weights g as a function of azimuth angle, element number and range gives us a new array factor formulation dependent on range as in 3-10. In the same study, it's also proposed that the computation of both r_n and r_{nfar} where r_n is actual target range and r_{nfar} should be selected as an arbitrary far range where one can obtain a desirable beam pattern [17]. Whereas in another study using the far-field phases to approximate to the most ideal beam pattern has been proposed to obtain an ideal far-field beam pattern [18]. The latter method has been chosen in our research to proceed, because focusing the beam to ideal far-field beam patterns would give us the means to compare our results to our findings from previous chapter. We already know that ideal phase shifts from a target is as shown in 3-14 from the first chapter. Therefore, using 3-7 the difference between the real phase shift and the ideal phase shift can be computed as shown in 3-15.

Shifting the phases this way is like digitally adjusting the locations of the array elements such that they would look like a lens (hence the sometimes referral of the method as lensing). In fact, it affects the effective Fraunhofer distance as well. With the introduction of the added phase shifts, the array factor now transforms into a more desirable far-field pattern (at the range r_{nfar}). The drawback is that this method requires the target range to be pre-known. But assuming a collocated MIMO array (a MIMO array where transmit and receive arrays are spatially located very close to each other that return trip time is close enough with respect to a target)[1].

In summary, proposed method is; compensation of the signal phases to match the signal phases coming from far field. After the received signal phases are adjusted to far-field phases, beam steering and all other beamforming techniques can still be used as is, because this process is an LTI (Linear, Time-invariant) process. For a 15 element phased array, the computed results of this processing can be seen below in Figure 3-3. The beam patterns were computed at far-field and at 20λ range (where the beam pattern breaks completely) with and without near-field focusing methods for comparison.

In Figure 3-3 the beam pattern at 20λ range without near-field focusing (blue legend) can be seen as it was as in Figure 3-2. Sidelobe levels are as high as the main lobe level, and the directive gain has been lost about 3 dB. The beam pattern at 20λ range with near-field focusing (red legend) can be seen as it has the shape of an almost ideal Sinc function. The ideal far-field beam pattern (green legend) can be seen as a Sinc function as obtained previously via computations.

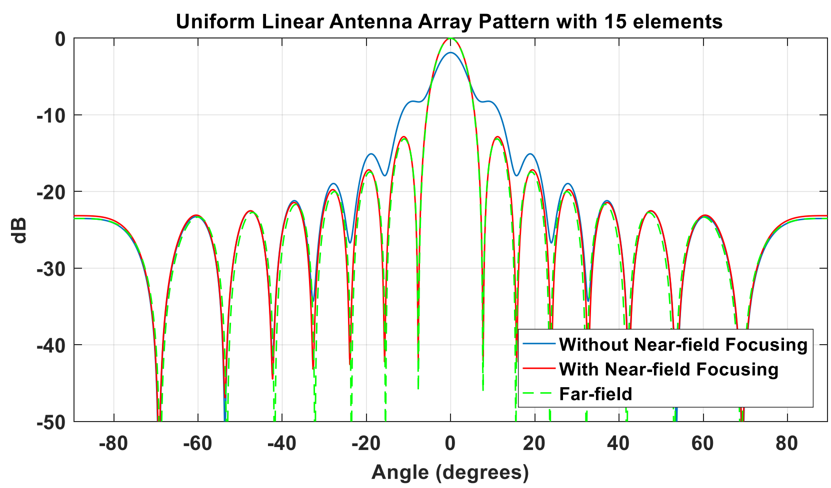


Figure 3-3 Phased array beam-patterns at far-field and at near-field with and without near-field focusing at 20λ range

3.2.4 **MIMO Array Focusing Techniques**

As explained in the previous chapter, due to the orthogonality of the transmitted waves in a MIMO radar, a linear MIMO radar's array-factor can be rewritten as the multiplication of its transmit and receive array factors as explained in the previous chapter. This property also allows virtual array element signals to be steered with receive array steering weights, and then the transmit array steering weights separately. And this is equivalent to computing a generic MIMO steering vector based on virtual element positions, where multiplication of a virtual element signal with this steering vector produces a phase shift equivalent to steering the virtual array directly to a specific angle. After the steering vectors are computed, they are then multiplied with the signals and all signals are summed to produce a single beam looking at a specific angle. Repeating this process for all angles of interests then yields a range-angle map showing the reflection intensity at different range-angle pairs. The summary of this approach can be seen again in Figure 2-5 as introduced in the previous chapter, repeated here as a basis for comparison with near-field processing.

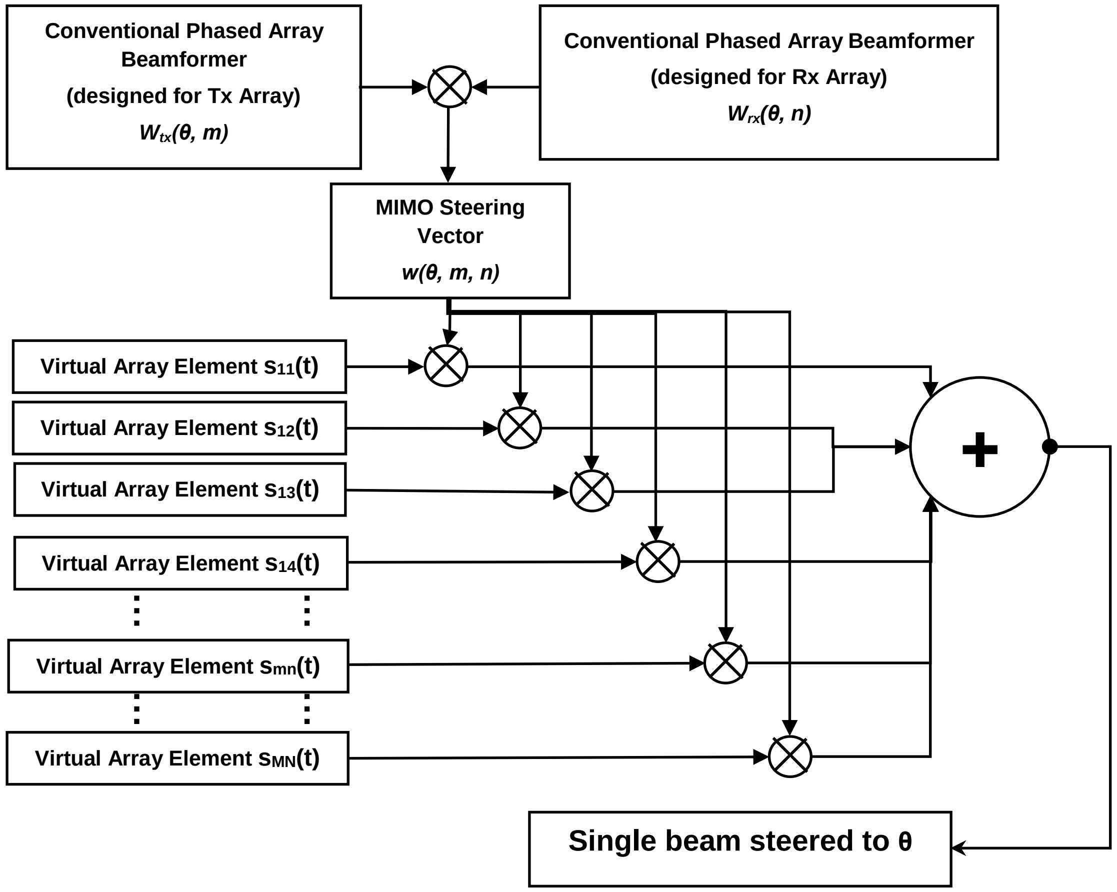


Figure 3-4 Block diagram for MIMO beamformer at far field

$$AF_{mimo} = \sum_{n=1}^{N} w_{ntx}(\theta) * e^{-jkr_n} \sum_{m=1}^{M} w_{nrx}(\theta) * e^{-jkr_m} \qquad 3\text{-}16$$

The MIMO array factor using this method of beam-steering can be seen in 3-16. Using the exact range r_n as derived above in 3-7, we can then attempt to numerically compute the beam patterns at various ranges, as performed for a uniform linear phased array. To be able to do this we would have to compute the exact ranges from each transmitter and receiver antenna to an arbitrary target with radial coordinates (r, θ).

$$r_m = \sqrt{r^2 + \sin(\theta)\,(M - 2m + 1)\Delta x + \frac{(M - 2m + 1)^2 (\Delta x_{tx})^2}{4}} \tag{3-17}$$

$$r_n = \sqrt{r^2 + \sin(\theta)\,(N - 2n + 1)\Delta x_{rx} + \frac{(N - 2n + 1)^2 (\Delta x_{rx})^2}{4}} \tag{3-18}$$

Since the transmit array and receive array are nothing but ordinary linear arrays, the computation of exact ranges to a target from transmit and receive array elements separately can be done simply as it was done for a phased array. So, the ranges from transmit and receive arrays r_n and r_m can be written as 3-17 and 3-18 using 3-7. Note that w_{ntx} and w_{nrx} in 3-16 are the beam steering weights.

In Figure 3-5 below, numerically computed beam patterns of a 3x5 MIMO array with 3-element coarse and 5-element fine array at various ranges can be seen. The Fraunhofer distance for such an array is 50λ where λ is the wavelength. At a reasonable far-range such as 10000λ (blue legend), it can be seen in the blue legend of Figure 3-5 that the beam pattern is looks like a Sinc function. 98λ range was not taken into account this time since this range is twice the Fraunhofer distance now. range At 50λ range (red legend), it can be seen that even though the shape of the beam is still very similar to a Sinc function, the nulls of the beam start to rise up. This is because in conventional MIMO beamforming nulls of the fine array cancels out the grating lobes of the coarse array and the non-ideal nulls of the fine array cause the rise-up in MIMO beam pattern as well. But an important detail is that these effects now only start to show up at only half the range as it was in a phased array. This is due to the fact that Fraunhofer distance has dropped due to the virtual array effect. Fraunhofer distance is determined by the maximum physical length of an aperture, when MIMO arrays are used, they form a virtual aperture which is the equivalent of a bigger aperture without actually increasing the physical aperture size. So, already beamforming at nearer ranges is more feasible. This shows an already apparent advantage of using MIMO arrays at near-field over phased arrays. At 20λ range (orange legend) it can be observed that the nulls quickly start to disappear along with a slight increase in sidelobe levels and slight decrease in directive gain. At 10λ range (magenta legend) it can be seen that the main lobe is only as strong as the first sidelobes (or vice versa), and that the directive gain loss is so high that the beam pattern is now currently unusable. Finally, at 5λ range (green legend), it can be observed that the beam pattern is not at all usable; main lobe level is even lower than the first sidelobes (so much that it's technically wrong to call it main lobe). To put these ranges into context, the wavelength that's been used in previous chapter's experiments was 8.6 cm. The Fraunhofer distance, calculated with an aperture length of ~43 cm for this wavelength is ~4.3 meters, and 20λ range where nulls are significantly lost is 1.7 meters. An equivalent phased array would have a ~60 cm aperture length and the Fraunhofer distance would be ~8.4 meters.

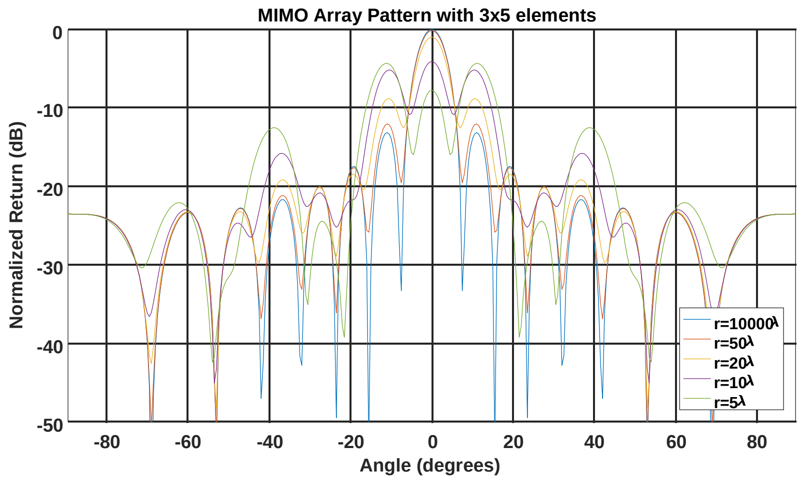


Figure 3-5 MIMO array factor in near field at various ranges (in terms of electrical lengths)

The previously proposed near-field focusing method can now be applied separately to receive and transmit arrays, simply by introducing near-field focusing weights separately to individual array factors. Using the equations derived previously for a uniform linear array, separate complex weights to compensate the near field effects of transmit and receive arrays can be computed. These separately computed complex weights then compensate for the near-field effects of the arrays on an individual array basis, or for the virtual array combined.

$$r_m - r_{mfar} = \sqrt{r^2 + \Delta x_{tx}\sin(\theta)\,(M - 2m + 1) + \frac{(M - 2m + 1)^2(\Delta x_{tx})^2}{4}} - r + \left(\frac{M - 2m + 1}{2}\right)\Delta x_{tx} sin(\theta) \qquad \text{3-19}$$

$$r_n - r_{nfar} = \sqrt{r^2 + \Delta x_{rx}\sin(\theta)\,(N - 2n + 1) + \frac{(N - 2n + 1)^2(\Delta x_{rx})^2}{4}} - r + \left(\frac{N - 2n + 1}{2}\right)\Delta x_{rx} sin(\theta) \qquad \text{3-20}$$

Using 3-15 the separate slant range differences for transmit and receive elements can be written as in 3-19 and 3-20. These range differences then would be used to compute phase differences to shift the received virtual element signal phases.

$$f_{tx}(\theta, m, r) = e^{-jkr_{mfar}} e^{jkr_m} = e^{jk(r_m - r_{mfar})} \quad \textit{3-21}$$

$$f_{rx}(\theta, n, r) = e^{-jkr_{far}} e^{jkr_n} = e^{jk(r_n - r_{nfar})} \quad \textit{3-22}$$

And then the filters can be simply written separately using 3-10 as a reference as shown in 3-21 and 3-22. These filters compute the necessary phase shifts for transmit and receive array separately, and when combined they produce the necessary phase shifts to focus the virtual array.

$$AF_{mimo} = \sum_{n=1}^{N} \left(w_{ntx}(\theta) * f_{tx}(\theta, n, r) * e^{jkr_n}\right) \sum_{m=1}^{M} \left(w_{nrx}(\theta) * f_{rx}(\theta, m, r) * e^{jkr_m}\right) \quad \textit{3-23}$$

With the introduction of near-field focusing weights the MIMO array factor now takes the formulation as shown in 3-23, similar to 3-12.

The now modified signal processing system block diagram can be seen in Figure 3-6 compared to non-near-field focused system block diagram in Figure 2-5. First, near-field focusing phase shifts are computed for virtual array elements by computing the near-field focusing shifts for transmit and receive array separately and multiplying them. Note that these phase shifts are computed for each range bin separately, effectively having a different “lens” for each range bin, therefore focusing each range-angle pair. Then, the computed phase shifts are multiplied with the received signals to shift the received signal phases to ideal far-field phases. Then, beam steering phase shifts are applied to signals, and then summed to obtain a single beam steered to a specific angle.

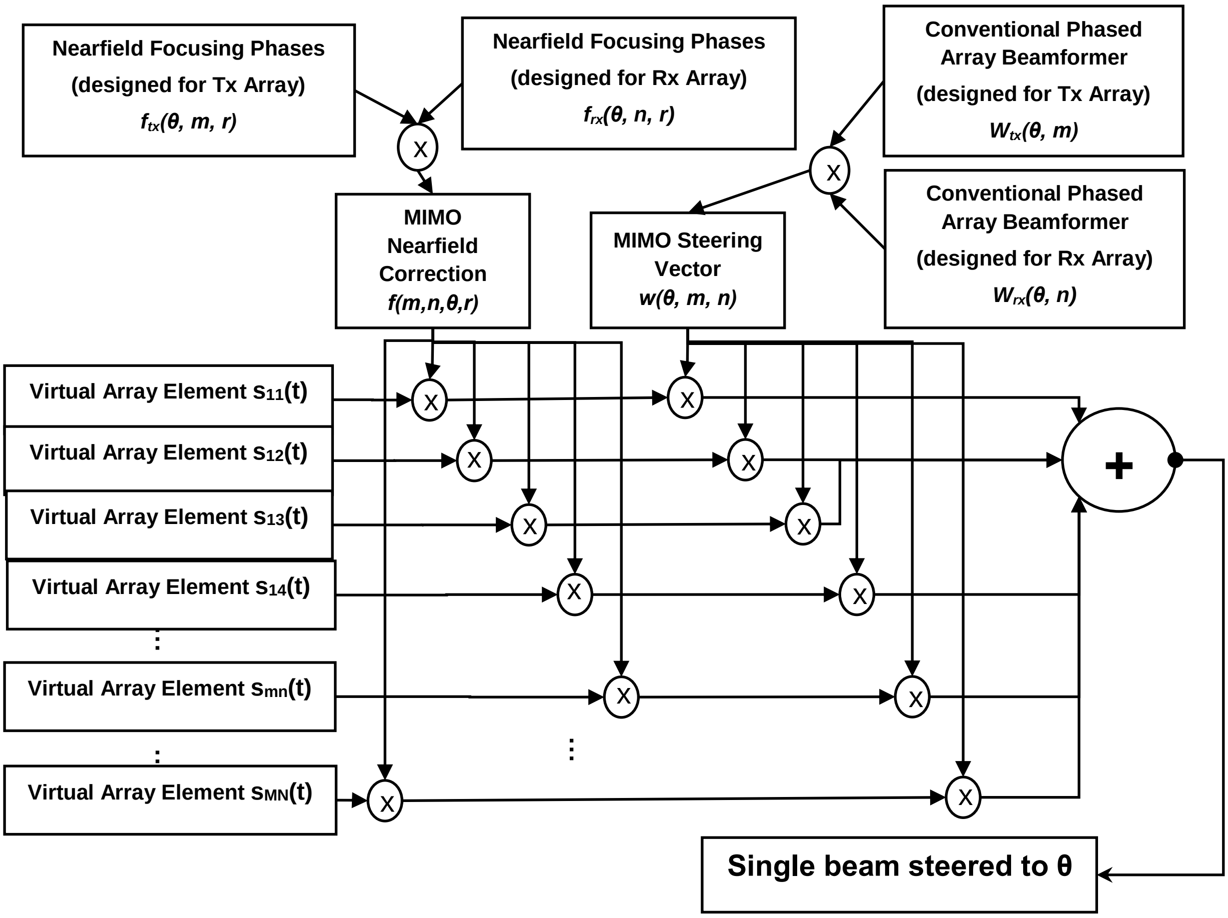


Figure 3-6 Block diagram for MIMO beamformer at near-field

Repeating this process for all range/angle combinations then yields a range-angle map showing the reflection intensity at different range-angle pairs. Below in Figure 3-7 the beam patterns of the same 3x5 MIMO array computed with and without near-field focusing methods at 10λ range (where the beam pattern is completely unusable) and the ideal far-field beam patterns can be seen and compared.

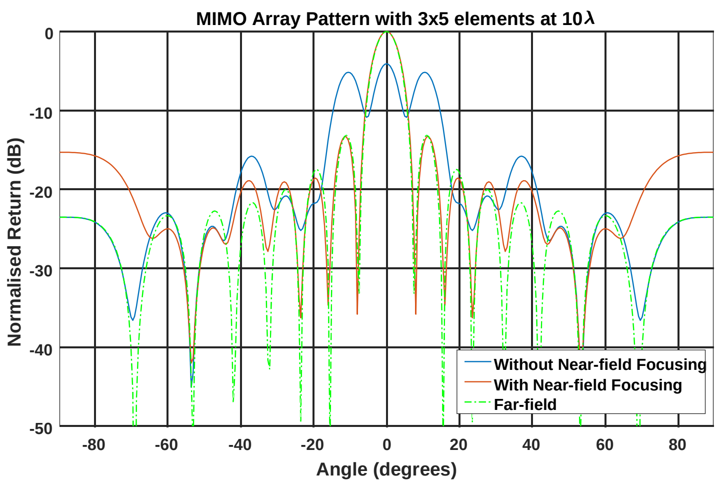


Figure 3-7 Far-field and near-field beam-patterns with and without near-field focusing

In Figure 3-7 the beam pattern at 10λ range without near-field focusing (blue legend) can be seen as it was as in Figure 3-5. Sidelobe levels are as high as the main lobe level, and the directive gain has been lost about 5 dB. The beam pattern at 10λ range with near-field focusing (red legend) can be seen as it has the shape of an almost perfect Sinc function apart from the relatively higher sidelobes outside of -65 and 65 degrees. The ideal far-field beam pattern (green legend) can be seen as a Sinc function as obtained previously via computations. The ideal far-field beam pattern makes it easier to observe the difference of relatively higher sidelobe levels in near-field focused beam pattern. However, this slight increase in sidelobe levels are negligible since the level is still under the first sidelobe level of -13 dB.

3.3 SIMULATIONS

As mentioned in the previous chapter a MIMO radar simulator has been coded to approximate our laboratory experiment environment. The simulation programme works with ray tracing in 3-dimensional coordinate space, therefore it generates received signals as if waves were actually fired from our transmitters, bounced off of point targets and picked up by receivers. The simulation programme simply assumes point targets at any distance. In addition to our MIMO simulator, a phased-array simulator programme was made in order to be able to simulate the already existing advantages of using MIMO arrays in near-field over phased-arrays. This previously mentioned advantage is the decreased Fraunhofer distance of the array, allowing for already more focused beams even without near-field focusing. Note that also as shown before, this has its limitations, which is why near-field focusing is required.

For the set of simulations to back up experimental results, same parameters as previous chapter were used which can be found in Table 2-1 since they are the closest parameters that was used to conduct our MIMO experiments (the experiment parameters can be seen in Table 2-3).

3.4 EXPERIMENTAL RESULTS

In order to be able to compare our results with previously obtained ones (see previous chapter), the experimental setup was kept the same as before. Only difference was the range that the targets were placed and the variation of scenarios. The summary of the experiment parameters can be seen again in Table 2-3. This time, experimental scenarios were designed to prove beamforming capabilities at various angles and –near-field- ranges and not to prove other MIMO capabilities such as multiple target resolution.

Prior to measurements with targets, recordings with the empty anechoic chamber were made and processed to form the corresponding range/angle maps. Those were then subtracted from any recording with a target, which allowed compressed echoes from the chamber itself and direct signal artefacts to be suppressed, hence allowing for a better assessment of beamforming performance.

This method is good for proving beamforming capabilities however it's not practical due to at least two major reasons. First is that, it's simply impossible to have a "background measurement" in real-life environments. Second is, simple subtraction is not able to remove multi-path effects (such as ghost images), therefore it's likely to fail when multiple strong targets are present and/or multi-path effects are significant. Therefore this method is only used in this chapter to prove near-field beamforming capabilities with a single target present. However, solving issues of direct path transmission and other background artefacts is not in the scope of this research, therefore a more sophisticated approach was neither needed nor explored.

3.4.1 Scenarios and Results

Various experiments were performed where target(s) were placed about 1.2m away from the radar and changing azimuth angles, to identify the angular range over which the MIMO array can perform. Below are the 3 scenarios simulated with various configurations and experimented with a 3x5 MIMO radar. The scenarios are presented with from-the-top sketches and photos taken inside the anechoic chamber during the experiments. From-the-top sketches are also superimposed with transparent version of from-the-top photos (note that some photos are stretched to match plot dimensions). Next to the top view sketches of setups are the photos taken inside the anechoic chamber during experiments. These scenarios are designed to check beamforming and target localization capabilities, azimuth resolution but not multiple target resolution capabilities.

Also below are the results of 3x5 MIMO radar experiments compared to 15-element phased array simulation, compared to 3x5 element MIMO simulation without nearfield focusing and finally compared to 3x5 MIMO simulation with nearfield focusing. Range-angle maps are presented and are accompanied with their azimuth cuts at target ranges. The experiments and simulations have been performed at approximately 0, 25, and 30 degrees.

The results are presented and compared in a logical order showing the improvements of using MIMO radar compared to phased arrays, and then showing the improvements of our near-field focusing methods compared to not doing near-field focusing, and finally experimental results are presented to show our methods work in practice. Experimental results will also be compared to phased-array simulations to emphasize overall improvement of using MIMO arrays over using conventional phased arrays. An extra benefit in this approach is that since artefacts in such an environment are difficult to compute, a comparison of results through various array configurations and processing schemes can help visualise the kind of results they can produce as well as their relative merits.

Single Target at 0 Degrees

In Figure 3-8 the over the top sketch of the first scenario can be seen superimposed with a stretched-to-fit top view photo of the experimental setup. A target is placed at about 1.15 metres slant range with about 0.5 degree angle. The photo of the setup on its own can be seen in Figure 3-9. This scenario was designed to verify single target detection capabilities at near-field at near zero angle (beamwidth, sidelobe levels, etc.).

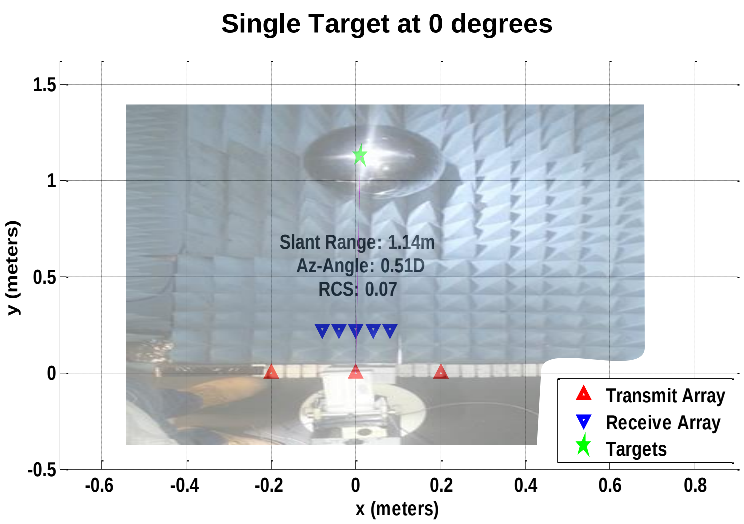


Figure 3-8 Single target at 0 degrees, scenario sketch of the experiment and simulation

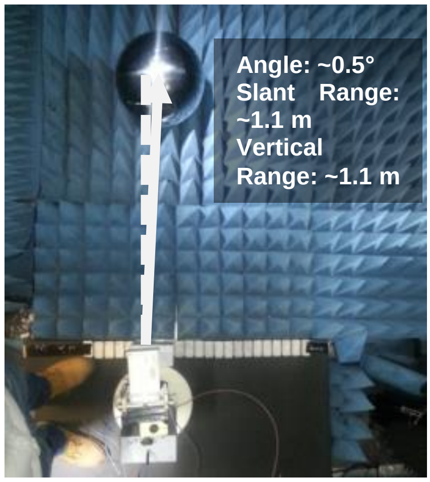


Figure 3-9 Single target at 0 degrees, top-view photo of the experiment

Below in Figure 3-10 the simulated range-angle map of a target at 0.5 degrees with a phased array, accompanied with its azimuth cut in Figure 3-11 at the target range. The target can be located at 0 degrees and at 1.15 metres. In the azimuth cut, the beam pattern can be observed at the target range. The beam structure at this range is heavily distorted and not a desired beam pattern as expected from a phased array. The beamwidth measurement is not possible because first and even second nulls are lost. First sidelobe level is measured as -1.75 dB, however even with such a high value this value is not an indicator of anything since most of the nulls of the beam pattern are lost. The shape of the sidelobe structure is as expected from a phased array operating at this range; structure shows a decreasing pattern as it gets further from the target, but first and second nulls are lost.

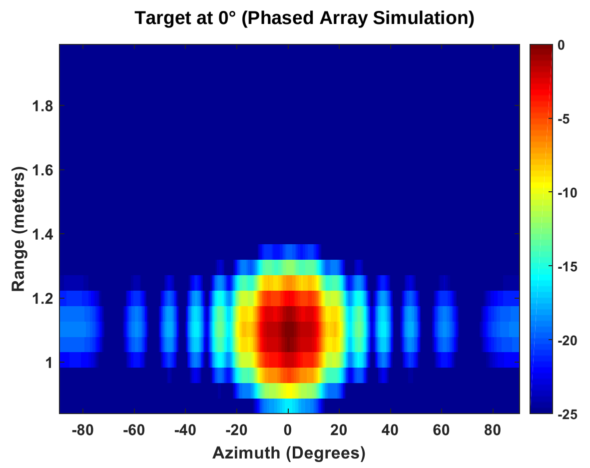


Figure 3-10 Range-angle map obtained from simulating a 15-element phased array with a near-field target at 0 degrees

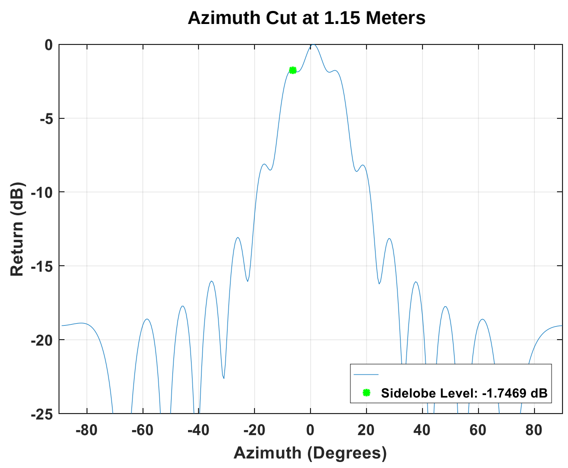


Figure 3-11 Azimuth cut of range-angle map obtained from simulating a 15-element phased array with a near-field target at 0 degrees

Below in Figure 3-12 the simulated range-angle map of a target at 0.5 degrees with a MIMO array without near-field focusing, accompanied with its azimuth cut in Figure 3-13 at the target range. The target can be located at 0 degrees and at 1.15 metres, but alongside it can be observed relatively high sidelobe levels. In the azimuth cut, the beam pattern can be observed at the target range. The beam structure at this range is also heavily distorted and still not a desired beam pattern. Arguably though, this is a better pattern compared to the phased array result, since first nulls are not completely lost. The beamwidth is measured to be 6.46 degrees which is in the 10% vicinity of the expected beamwidth of 7.15 degrees. First sidelobe level is measured as -4.09 dB, however –like before- even with such a high value this value is not an indicator of anything since first nulls of the beam pattern are not low enough. The shape of the sidelobe structure is as expected from a MIMO array operating at this range; structure shows a decreasing pattern as it gets further from the target, but presence of high first sidelobe levels are observed because of failure in beamforming at such short ranges.

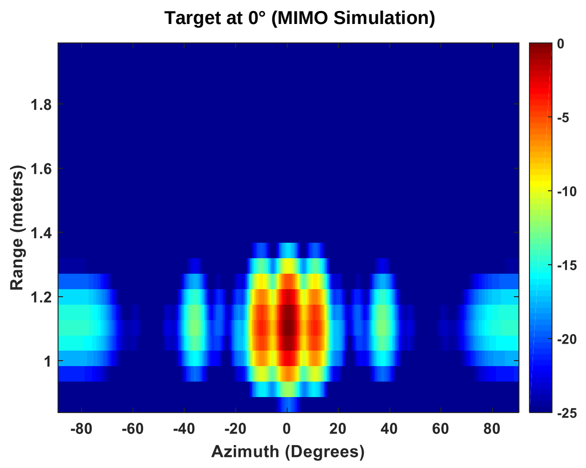


Figure 3-12 Range-angle map obtained from simulating a 3x5 MIMO array with a near-field target at 0 degrees without near-field focusing

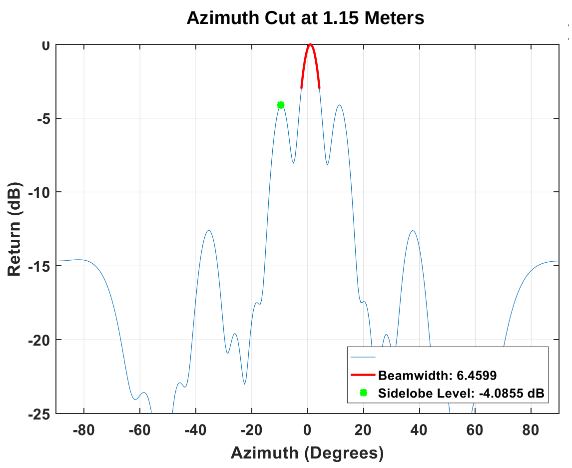


Figure 3-13 Azimuth cut of range-angle map obtained from simulating a 3x5 MIMO array with a near-field target at 0 degrees without near-field focusing

Below in Figure 3-14 the simulated range-angle map of a target at 0.5 degrees with a MIMO array with near-field focusing, accompanied with its azimuth cut in Figure 3-15 at the target range. The target can be located at 0 degrees and at 1.15 metres. In the azimuth cut, the beam pattern can be observed at the target range. The beam structure at this range is not distorted at all and looks like a Sinc function one would obtain at far-field. The beamwidth is measured to be 6.92 degrees which is in the 10% vicinity of the expected beamwidth of 7.15 degrees. First sidelobe level is measured as -13.63 dB. The shape of the sidelobe structure is as expected from a MIMO array operating at this range; structure shows a decreasing pattern as it gets further from the target, and due to the near-field focusing it is very similar to a far-field beam pattern. This simulation clearly shows the effect of near-field focusing in such close ranges.

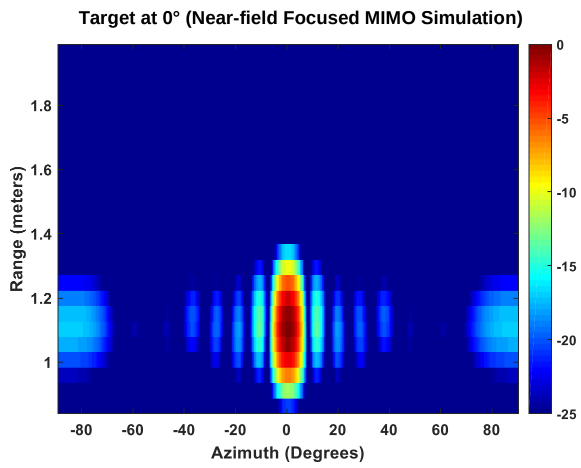


Figure 3-14 Range-angle map obtained from simulating a 3x5 MIMO array with a near-field target at 0 degrees with near-field focusing

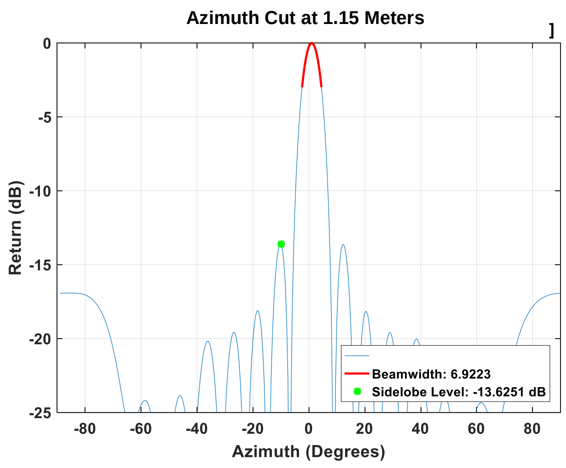


Figure 3-15 Azimuth cut of range-angle map obtained from simulating a 3x5 MIMO array with a near-field target at 0 degrees with near-field focusing

Below in Figure 3-16 is the experimentally acquired range-angle map of a target at ~0 degrees with a MIMO array with near-field focusing, accompanied with its azimuth cut in Figure 3-17 at the target range. The target can be located at 0 degrees and at 1.15 metres. In the azimuth cut, the beam pattern can be observed at the target range. The beam structure at this range is very similar to the simulated result with minor differences. The beamwidth is measured to be 6.83 degrees which is in the 10% vicinity of the expected beamwidth of 7.15 degrees. First sidelobe level is measured as -11.67 dB, only 2 dB less than the simulated result. The shape of the sidelobe structure is as expected from a MIMO array operating at this range; structure shows a decreasing pattern as it gets further from the target, and due to the near-field focusing it is very similar to a far-field beam pattern, albeit with some asymmetry in the sidelobes.

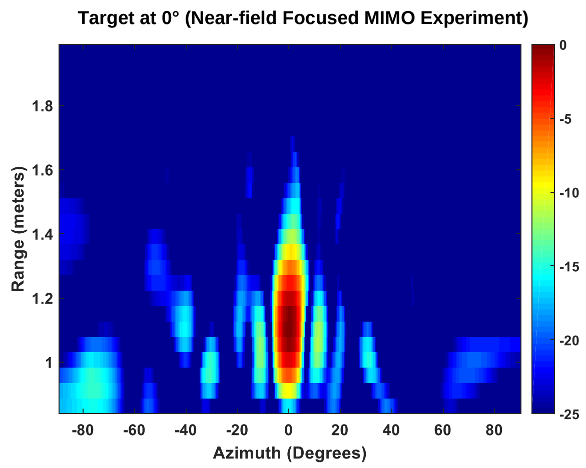


Figure 3-16 Range-angle map obtained from the experimental 3x5 MIMO array setup with a near-field target at 0 degrees with near-field focusing

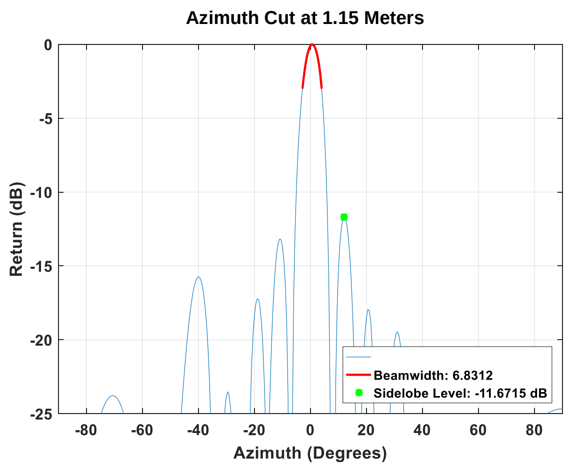


Figure 3-17 Azimuth cut of range-angle map obtained from the experimental 3x5 MIMO array setup with a near-field target at 0 degrees with near-field focusing

Single Target at 25 Degrees

In the Figure 3-18 over the top sketch of the first scenario can be seen superimposed with a stretched-to-fit top view photo of the experimental setup. A target is placed at about 1.1 metres slant range with about 25 degree angle. The photo of the setup on its own can be seen in Figure 3-19. This scenario was designed to verify single target detection capabilities at near-field at an angle (beamwidth, sidelobe levels, etc.).

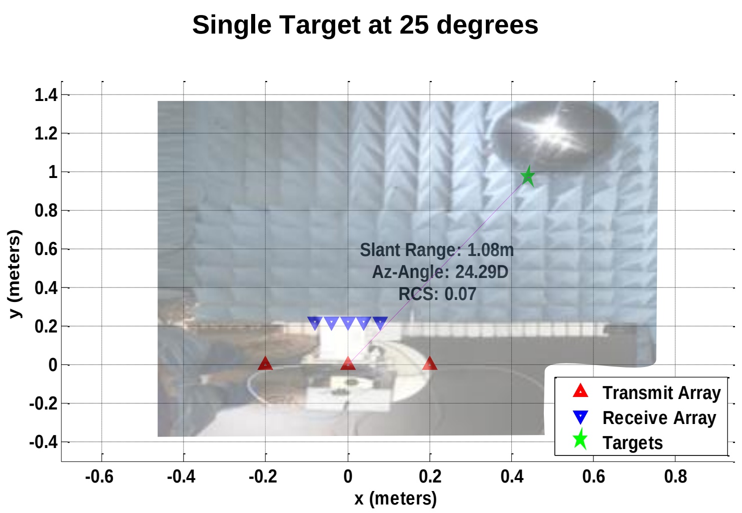


Figure 3-18 Single target at 25 degrees, scenario sketch of the experiment and simulation

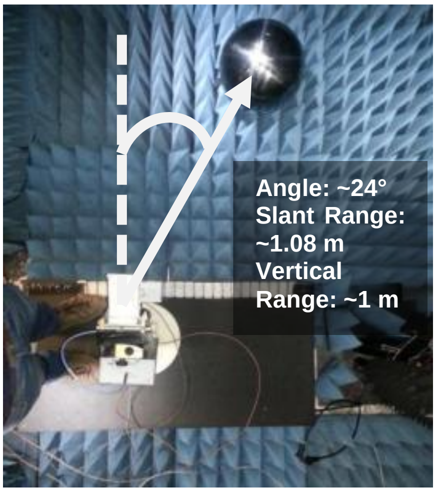


Figure 3-19 Single target at 25 degrees, top-view photo of the experiment

Below in Figure 3-20 the simulated range-angle map of a target at 25 degrees with a phased array, accompanied with its azimuth cut in Figure 3-21 at the target range. The target can almost be located at 23 degrees and at 1.06 metres, which is off by a couple degrees with respect to the simulation. In the azimuth cut, the beam pattern can be observed at the target range. The beam structure at this range is heavily distorted and not a desired beam pattern as expected from a phased array. The beamwidth measurement is somewhat possible and 3dB beamwidth can be measured as 12.60 degrees from the azimuth cut, but noting the first and even second nulls are lost.

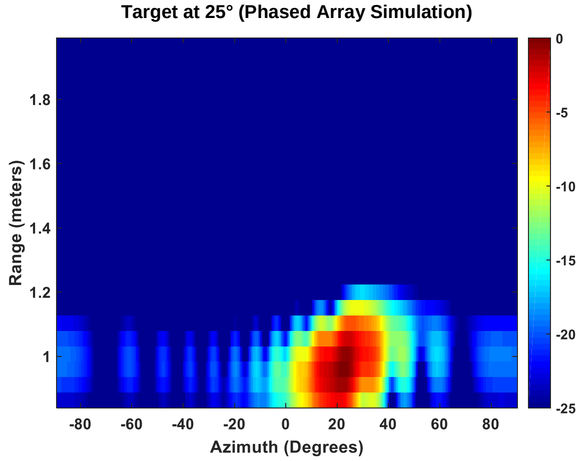


Figure 3-20 Range-angle map obtained from simulating a 15-element phased array with a near-field target at 25 degrees

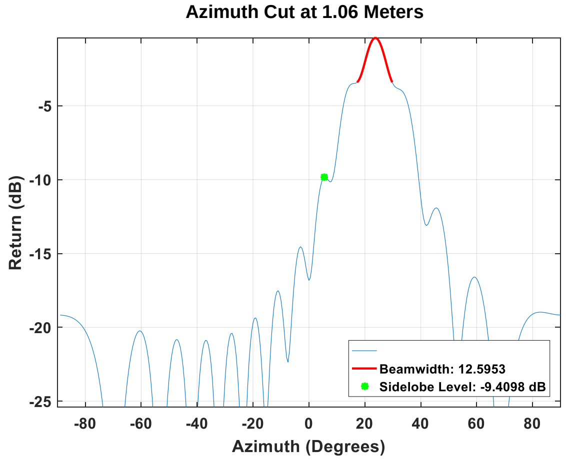


Figure 3-21 Azimuth cut of range-angle map obtained from simulating a 15-element phased array with a near-field target at 25 degrees

Below in Figure 3-22 the simulated range-angle map of a target at 25 degrees with a MIMO array without near-field focusing, accompanied with its azimuth cut in Figure 3-23 at the target range. The target can be located at 23.5 degrees and at 1.10 metres, which is 1 degree off compared to the simulation scenario. Alongside the main lobe are again relatively high sidelobe levels. In the azimuth cut, the beam pattern can be observed at the target range. The beam structure at this range is also heavily distorted and still not a desired beam pattern. Arguably though, this is a slightly better pattern compared to the phased array result, since first nulls are not completely lost. The beamwidth is measured to be 7.57 degrees which is in the 10% vicinity of the expected beamwidth of 7.85 degrees.

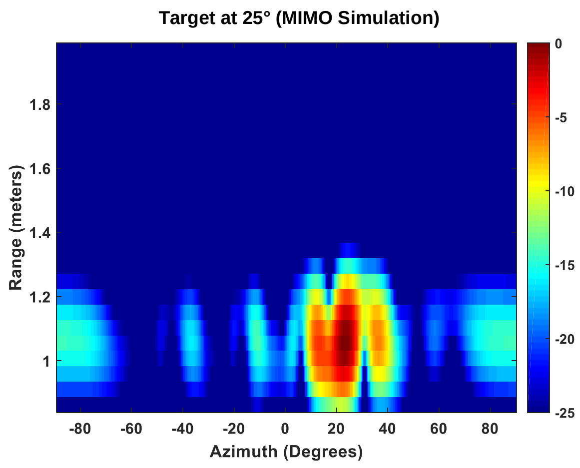


Figure 3-22 Range-angle map obtained from simulating a 3x5 MIMO array with a near-field target at 25 degrees without near-field focusing

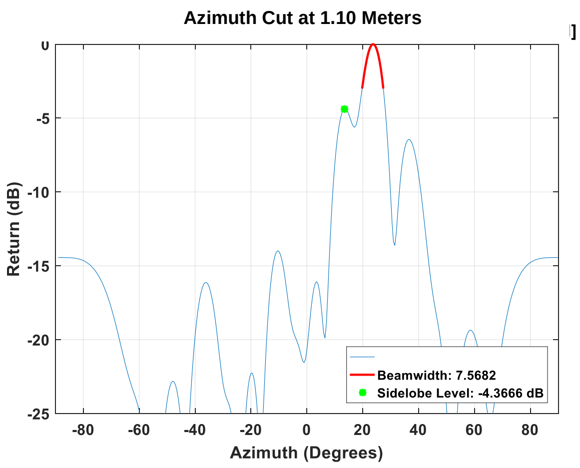


Figure 3-23 Azimuth cut of range-angle map obtained from simulating a 3x5 MIMO array with a near-field target at 25 degrees without near-field focusing

Below in Figure 3-24 the simulated range-angle map of a target at 25 degrees with a MIMO array with near-field focusing, accompanied with its azimuth cut in Figure 3-25 at the target range. The target can be located at 24.5 degrees and at 1.10 metres. In the azimuth cut, the beam pattern can be observed at the target range. The beam structure at this range resembles a Sinc function. The beamwidth is measured to be 7.66 degrees which is in the 10% vicinity of the expected beamwidth of 7.85 degrees. First sidelobe level is measured as -12.73 dB. The shape of the sidelobe structure is as expected from a MIMO array operating at this range; structure shows a decreasing pattern as it gets further from the target, and due to the near-field focusing it is very similar to a far-field beam pattern. This simulation shows the effect of near-field focusing in such close ranges.

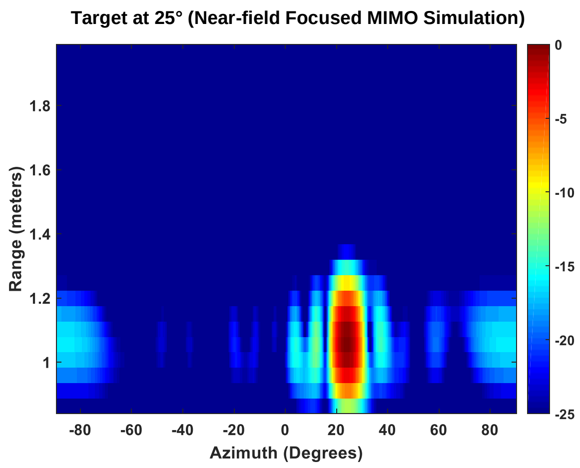


Figure 3-24 Range-angle map obtained from simulating a 3x5 MIMO array with a near-field target at 25 degrees with near-field focusing

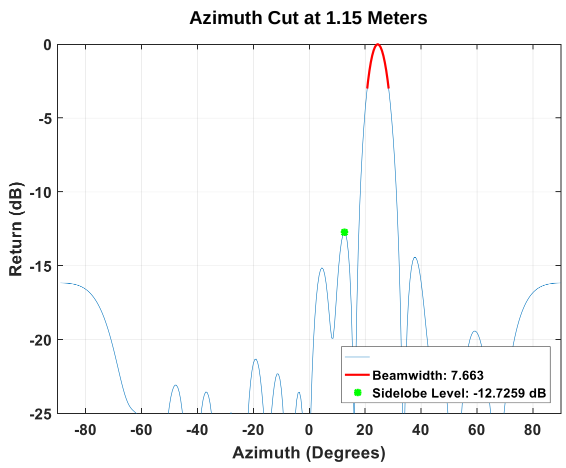


Figure 3-25 Azimuth cut of range-angle map obtained from simulating a 3x5 MIMO array with a near-field target at 25 degrees with near-field focusing

Below in Figure 3-26 is the experimentally acquired range-angle map of a target at ~24.5 degrees with a MIMO array with near-field focusing, accompanied with its azimuth cut in Figure 3-27 at the target range. The target can be located at 23.5 degrees and at 1.15 metres. In the azimuth cut, the beam pattern can be observed at the target range. The beamwidth is measured to be 7.95 degrees which is in the 10% vicinity of the expected beamwidth of 7.85 degrees. First sidelobe level is measured as -11.54 dB, only 1.2 dB less than the simulated result. The shape of the sidelobe structure is not identical to a Sinc function (see Figure 3-24). However, the 3-dB beamwidths do match simulation results while returns from the chamber are still visible (see returns at close ranges and negative angles for example) and hence could interfere with the response of the target.

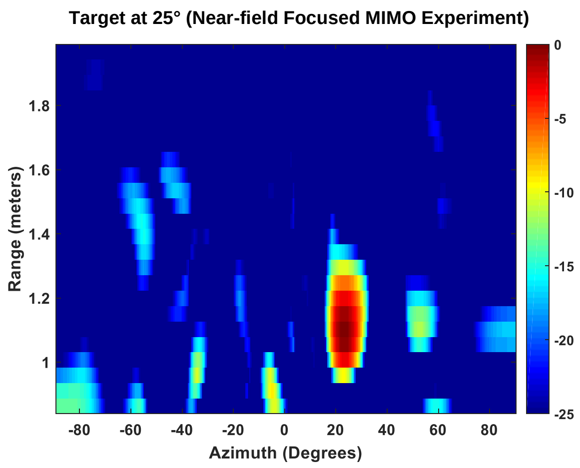


Figure 3-26 Range-angle map obtained from the experimental 3x5 MIMO array setup with a near-field target at 25 degrees with near-field focusing

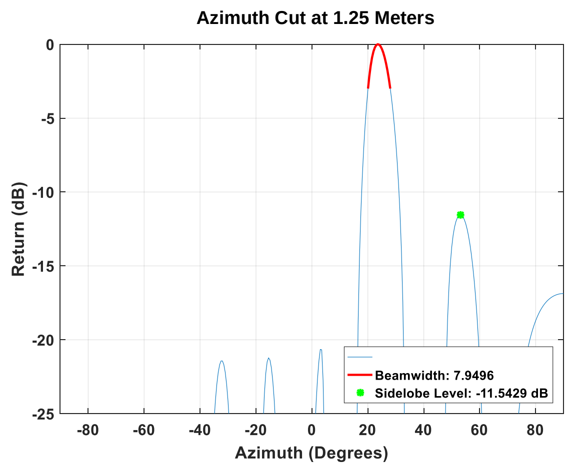


Figure 3-27 Azimuth cut of range-angle map obtained from the experimental 3x5 MIMO array setup with a near-field target at 25 degrees with near-field focusing

Single Target at 30 Degrees

In the Figure 3-28 over the top sketch of the first scenario can be seen superimposed with a stretched-to-fit top view photo of the experimental setup. A target is placed at about 1.3 metres slant range with about 30 degree angle. The photo of the setup on its own can be seen in Figure 3-29. This scenario was designed to verify single target detection capabilities at near-field at a high incident angle (beamwidth, sidelobe levels, etc.).

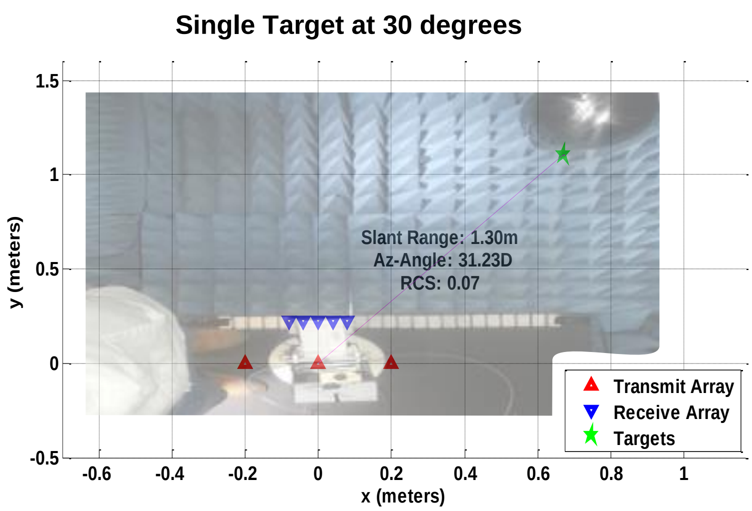


Figure 3-28 Single target at 30 degrees, scenario sketch of the experiment and simulations

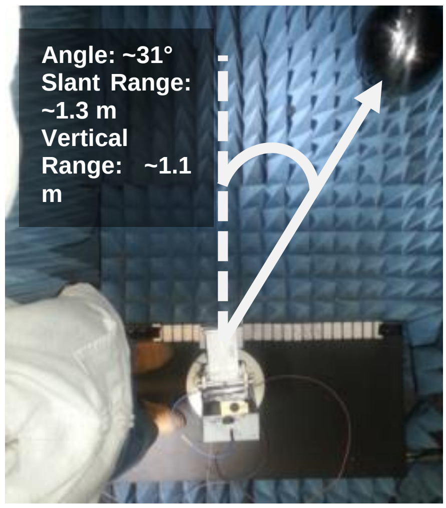


Figure 3-29 Single target at 30 degrees, top-view photo of the experiment

Below in Figure 3-30 the simulated range-angle map of a target at 31.23 degrees with a phased array, accompanied with its azimuth cut in Figure 3-31 at the target range. The target can be located at 30 degrees and at 1.25 metres. In the azimuth cut, the beam pattern can be observed at the target range.

The beamwidth measurement is somewhat possible and 3dB beamwidth can be measured as 9.38 degrees from the azimuth cut. But because first and even second nulls are lost, this value should not be taken as an indicator of performance.

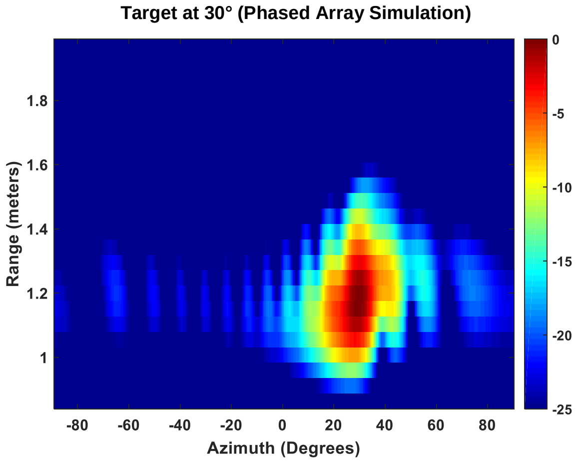


Figure 3-30 Range-angle map obtained from simulating a 15-element phased array with a near-field target at 30 degrees

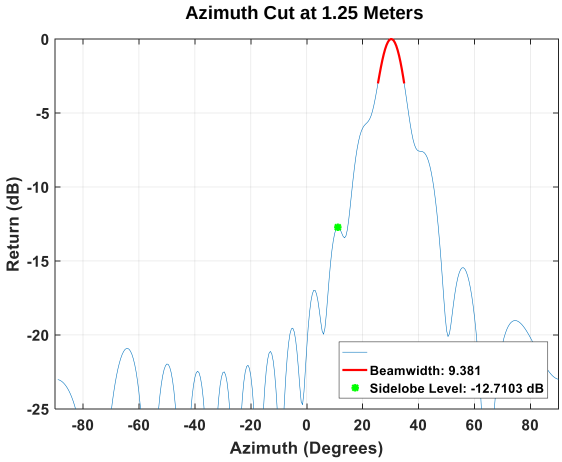


Figure 3-31 Azimuth cut of range-angle map obtained from simulating a 15-element phased array with a near-field target at 30 degrees

Below in Figure 3-32 the simulated range-angle map of a target at 31.23 degrees with a MIMO array without near-field focusing, accompanied with its azimuth cut in Figure 3-33 at the target range. The target can be located at 30.5 degrees and at 1.30 metres. Alongside the main lobe are again relatively high sidelobe levels. In the azimuth cut, the beam pattern can be observed at the target range. The beamwidth is measured to be 8.25 degrees which is in the 10% vicinity of the expected beamwidth of 8.35 degrees.

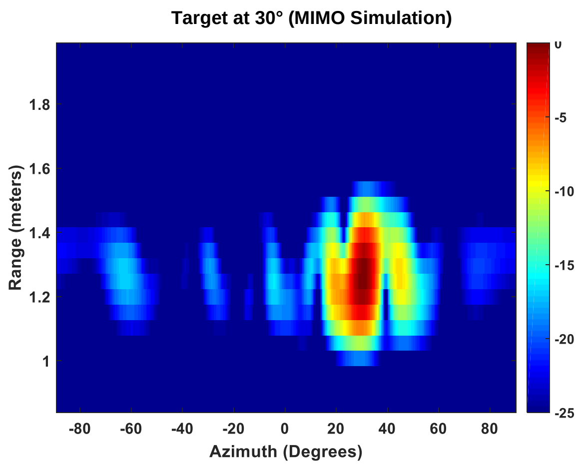


Figure 3-32 Range-angle map obtained from simulating a 3x5 MIMO array with a near-field target at 30 degrees without near-field focusing

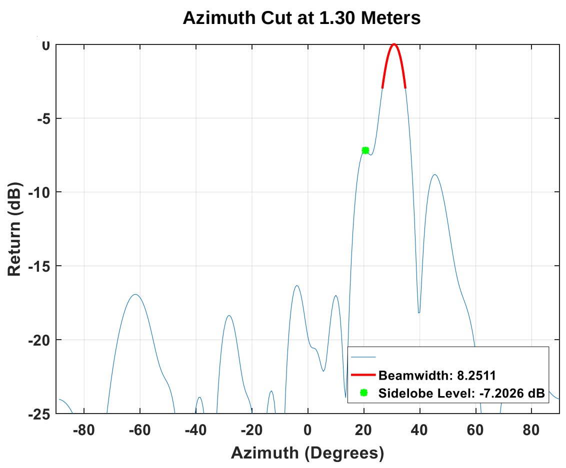


Figure 3-33 Azimuth cut of range-angle map obtained from simulating a 3x5 MIMO array with a near-field target at 30 degrees without near-field focusing

Below in Figure 3-34 the simulated range-angle map of a target at 31.23 degrees with a MIMO array with near-field focusing, accompanied with its azimuth cut in Figure 3-35 at the target range. The target can be located at 31.5 degrees and at 1.30 metres. In the azimuth cut, the beam pattern can be observed at the target range. The beam structure at this range is not distorted at all and looks like a Sinc function one would obtain at far-field. The beamwidth is measured to be 8.24 degrees which is in the 10% vicinity of the expected beamwidth of 8.35 degrees. First sidelobe level is measured as -13.14 dB. The shape of the sidelobe structure is as expected from a MIMO array operating at this range; structure shows a decreasing pattern as it gets further from the target, and due to the near-field focusing it is very similar to a far-field beam pattern. This simulation -again- clearly shows the effect of near-field focusing in such close ranges.

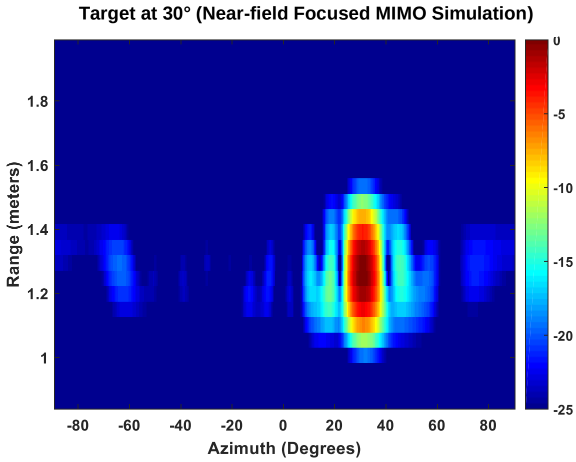


Figure 3-34 Range-angle map obtained from simulating a 3x5 MIMO array with a near-field target at 30 degrees with near-field focusing

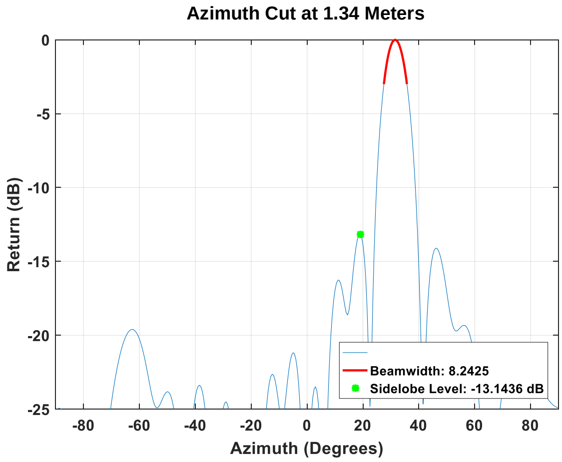


Figure 3-35 Azimuth cut of range-angle map obtained from simulating a 3x5 MIMO array with a near-field target at 30 degrees with near-field focusing

Below in Figure 3-36 is the experimentally acquired range-angle map of a target at ~31 degrees with a MIMO array with near-field focusing, accompanied with its azimuth cut in Figure 3-37 at the target range. The target can be located at 31 degrees and at 1.34 metres. In the azimuth cut, the beam pattern can be observed at the target range. The beamwidth is measured to be 7.78 degrees which is in the 10% vicinity of the expected beamwidth of 8.35 degrees. First sidelobe level is measured as -7.04 dB, 6 dB less than the simulated result. The relatively high sidelobe level can be explained due to target being at the edge of the 3-dB beamwidth of the transmitter antenna and that the background subtraction is not as effective as before. The shape of the sidelobe structure is not necessarily as expected from a MIMO array operating at this range; but the highest sidelobe level and the 3-dB beamwidth match the simulated results. Also important to note about the sidelobe structure is that; even though sidelobe levels do not completely match the simulation results –which could be due to various reasons- the locations (angles) of the most sidelobes that are within the dynamic range do match.

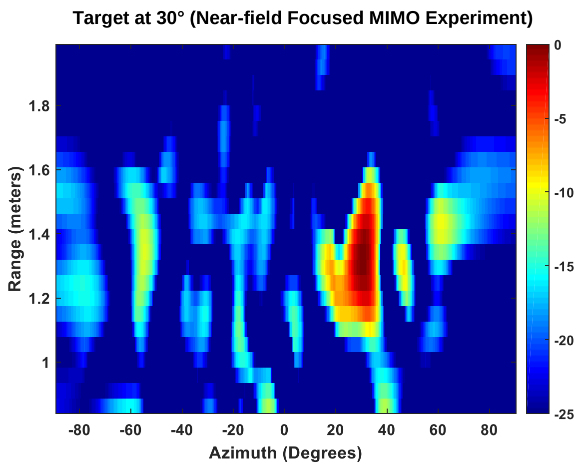


Figure 3-36 Range-angle map obtained from the experimental 3x5 MIMO array setup with a near-field target at 30 degrees with near-field focusing

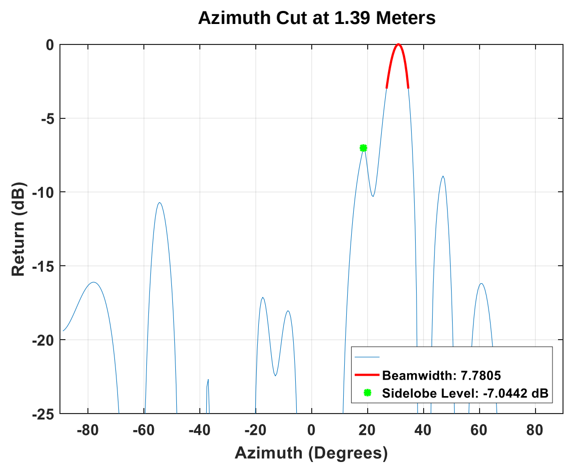


Figure 3-37 Azimuth cut of range-angle map obtained from the experimental 3x5 MIMO array setup with a near-field target at 30 degrees with near-field focusing

With the results above we confirm that we can acquire nearfield target images with MIMO radar configurations with special signal processing techniques.

3.5 Conclusions

The fundamental theory of near-field MIMO radar has been investigated. The geometry of near-field linear arrays has been introduced. And the derivation of the linear array factor in near-field without far-field approximations has been done. These formulations were then used to explain the behaviour of beam-patterns for linear arrays in near-field. Using the same formulations the behaviour of MIMO arrays in near-field were derived.

Existing methods of near-field focusing for phased arrays have been studied and the methods have been computationally verified at various ranges. These methods then were used to develop MIMO near-field focusing methods, and they have been computationally verified at various ranges to observe focusing effects.

MIMO near-field array factor computations and beamforming algorithms have been coded in MATLAB. In addition, a new simulation program was written to simulate phased arrays for comparison of MIMO arrays and phased arrays in near-field.

Same experimental setup from previous chapter was used for performing experiments in a controlled environment to verify experimental near-field beamforming capabilities using our MIMO setup. These experiments included scenarios containing a single target at near-field ranges and various angles to verify near-field focusing capabilities and to confirm expected performance such as 3-dB beamwidth and first-sidelobe level.

All experimental scenarios were simulated using the phased array and MIMO simulation programmes for comparison of experimental results, and for proof of improvement. It was found that the near-field MIMO array beamforming works well within expected parameters in a controlled environment, and that the experimental results match simulation results within nominal deviations up to a scan angle of 30 degrees. It was also confirmed that a near-field focused MIMO can perform significantly better than a non-near-field-focused MIMO, and/or a phased array. The obtained results confirmed understanding of near-field MIMO radar theory and the feasibility of our approach for the next steps, which is the design and development of a purpose specific MIMO technology demonstrator.

4 EXPERIMENTAL SYSTEM DESIGN & RESULTS

4.1 INTRODUCTION

In chapters 2 & 3, the MIMO theory and nearfield applications have been studied and experimented with. It was proven that the MIMO sensing theory works in lab environment, and signal processing algorithms for nearfield sensing were developed. These algorithms were then tested using simulations and later verified with experimental results. However, the experiments were performed with generic existing lab equipment, and not with purpose built hardware. To fully explore the capabilities of MIMO sensor arrays and to allow for further improvements, a sensor array more suitable for automotive purposes was custom built and tested via experiments.

For short-range sensing two types of MIMO sensor may be used; radar and ultrasonic. And it was desired to check the feasibility of both. An experimental radar system could be built with equipment readily available in the lab, but ultrasonic equipment needed to be custom-built. This chapter describes the ultrasonic equipment built to confirm MIMO principles explored in this thesis and the experimental confirmation of MIMO beamforming in near-field with this configuration. A total of 8 MIMO elements were assumed viable to demonstrate these principles. However, the techniques described here are applicable to any number, understanding that in an automotive environment this number should be generally small. In addition, ultrasonic sensing in automotive environment is desired because;

- there would be no need for frequency licensing,
- ultrasonic sensors are in general used for short-range sensing
- a number of ultrasonic sensors are already used in vehicles for a number of reasons (e.g. parking aid etc.), so they could be re-used.

In this chapter, how an ultrasonic demonstrator system design was made, how it's implemented and experimental results obtained with it will be explained. In first section the system specifications will be summarised. Next, the hardware and software system will be described with the support of system block diagrams. Later, hardware and software design and implementation details will be given. And finally, the experimental setup and the results obtained from various experimental scenarios will be presented. In seventh section a summary of this chapter will be given as conclusions.

4.2 REQUIREMENTS SPECIFICATIONS

The technology availability and cost aspects were discussed and, acoustic sensing was decided to be the better option due to various reasons. Firstly, using acoustic sensing would not require any frequency licensing which is a great deal from cost perspective. And since acoustic frequencies are much lower compared to common radio frequencies used for automotive sensing, it would also decrease the cost of components by an order. It is also easier to obtain smaller wavelengths, therefore higher range resolution could be obtained with acoustic frequencies (i.e. since range resolution depends on bandwidth and since it's easier to obtain higher bandwidths in higher frequencies which means in smaller wavelength, it's easier to obtain higher range resolution). The disadvantage of using acoustic sensing would be the loss of high detection ranges, however for the purposes of this research that was not necessary.

Sound waves are different than radio waves and therefore where it may be more advantageous to use them, they also have other disadvantages that can be associated. In its simplest physics of sound waves work by gas movement, which causes a temporal density change. Then the change in density is a change in pressure. And finally pressure differences generate gas motion[58]. The difference in such medium brings few key differences. First thing to notice is that there is no polarization property of sound waves, which can be of advantage since polarization of the sensors would not need to be considered. Another thing to notice would be that speed of sound is dependent on the density of the medium which can be affected easily by temperature, humidity and even wind. Potentially exhaust gases –if their density is significantly different to air- can cause lense effects and/or errors in range estimation on the formed images due to the change in sound speed. Other common weather features such as rain, ice, or similarly pebbles or spray would cause clutter, but this is not a problem specific to acoustics. Finally, there would be the issue of different absorption characteristics of ultrasonics. As known, attenuation of sound waves is much severe due to absorption compared to radio waves. However, due to the limited range this research was interested it was found not be an issue after checking with the literature[59]–[61]. Unfortunately a feasibility study of acoustics for coherent detection applications compared to blunt ranging -which is widely and commercially used- is not within the scope of this research and since the experiments were planned to be performed in a controlled environment, potential issues were not dwelled on.

Most acoustic components use 40 kHz centre frequency which is also the optimal frequency for sensing. Therefore a centre frequency of 40 kHz and approximately 8.4mm of wavelength was reached. The bandwidth of the signal was decided to be 4 kHz, yielding a theoretical range resolution of 4 cm without any windowing. This is also the maximum bandwidth that most available acoustic sensors can handle. Pulse width decision was initially set to an arbitrary length of 2.5ms. It was later increased during experiments in order to obtain higher SNR. It was decided to have a system with 20dB sensitivity in order to be able to have detection ranges up to 10 meters (the maximum dimension of our anechoic chamber) [59]. The final set of parameters will be explained in the experiments section. However, below in Table 4-1 below is the summary of the system characteristics based on equipment purchased.

Table 4-1 Summary of parameters for the system specifications

Parameter	**Value**	**Unit**
Number of transmitters	4	-
Number of receivers	4	
MIMO Beamwidth	~6.5	Degrees
Centre Frequency	40	kHz
Bandwidth	4	kHz
Wavelength	8.4	mm
Multiple Access Scheme	TDMA	-
Range resolution	0.04	m
PRI	>0.03	s
Waveform type	Up-chirp LFM	-

4.3 SYSTEM BLOCK DIAGRAM

The system can be generally depicted with a hardware system block diagram and a software system block diagram. Hardware system block diagram will explain the physical connections, clock management, and data flow through libraries and high-level software packages used. Software system block diagram explains data processing blocks such as low-pass and bandpass filters, matched filters, delay blocks for time correction, coherent summation, phase calibration, nearfield focusing, and tapering.

4.3.1 Hardware System Block Diagram

The hardware system is formed of a host PC, a data acquisition board, an Arduino board, receiver and transmit modules, and a custom housing for the transmitter and receiver boards. Below in in Figure 4-1 is a sketch of the hardware system block diagram. Host PC runs a Windows operating system and contains MATLAB for data processing and visualisation of outputs. It also contains the InstaCal libraries which enables the real time data transfer into MATLAB from the data acquisition board via a USB link. Due to the design of the DAQ (data acquisition board), we decided to feed an external stable clock source to the ADCs and DACs. This clock is generated and fed by an Arduino board, which is externally powered, so that the clock doesn't lose coherency between measurements. This means even if the power to the internal circuitry of the DAQ were to be cut off between transmissions during TDMA scheme, since the Arduino board would still be running, the clock source would not lose its coherency. The DAQ's outputs and inputs are connected to individual transmitter and receiver modules. These modules have analogue amplification and filtering circuitry on them and they have the ultrasonic sensors attached to them. The reason to have individual PCBs for each sensor was to be able to rapidly change MIMO element spacings at will.

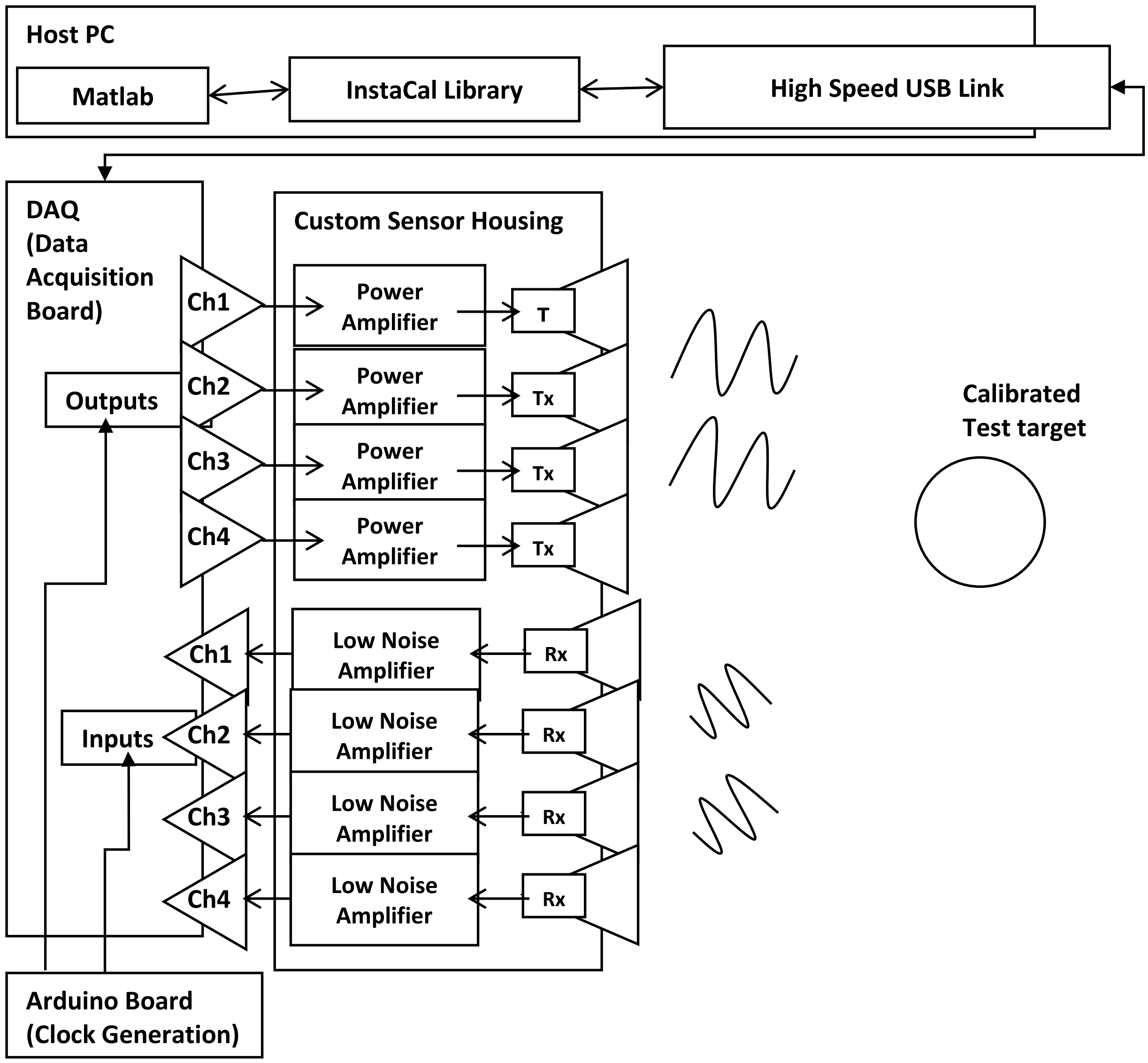


Figure 4-1 Hardware system block diagram of the technology demonstrator

Although the planned hardware consists of containing all sensors, due to physical limitations only one receiver module was used during the testing & experimentation phase. The physical limitations were that the receiver sensor size was too big to place the sensor with the required element spacing. The sensors would have to be either smaller, or would have to overlap somehow. Instead one receiver module was rotated between measurements and the data from these measurements were merged to virtually create the same output as the original hardware. Since the planned multiple access scheme was TDMA, this has caused only minimal change for the testing & experimentation phase.

4.3.2 Software System Block Diagram

The inner workings of the system can be described with a software system block diagram which is more detailed and can be used to port the system to different hardware. Most of the processing was implemented in MATLAB due to its high-level functions. Since this is firstly a proof of concept, reliability and stability was put to first priority rather than optimising computation time, portability and compatibility. The optimisation algorithms that are used to overcome common problems are not shown in these diagrams. To mention briefly, these algorithms are

- Pre-filtering of received signals (bandpass)
- Time-alignment of received signals by using the mutual induction of sensors
- Coherent summation of multiple frames to increase SNR
- Phase calibration of virtual element signals via use of a calibrated reference target

4.4 HARDWARE DESIGN

In this section, a more detailed look into the hardware design will be given. This will include the transmitter and receiver modules and the data acquisition board.

4.4.1 Data Acquisition

The data acquisition board was a Measurement Computing model USB-2537. A photo of this board can be seen in Figure 4-2. The key feature of this data acquisition board is that it has 4 output each of which can run at 1000 kilosamples/sec and 64 inputs which can run at a combined 1000 kilosamples/sec. So, with 4 inputs that would mean 250 kilosamples/sec for each channel. As explained before, we have decided to only use 4 outputs and 4 inputs.

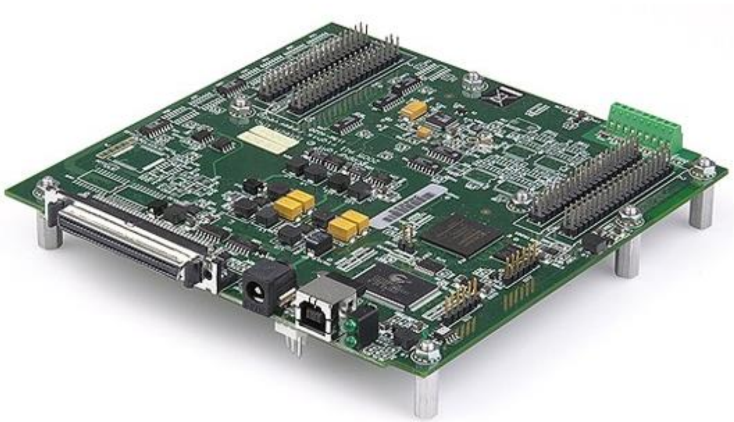


Figure 4-2 Measurement Computing Data Acquisition Board Model USB-2500 Series

The data acquisition from the board can be done real-time via a USB 2.0 link with the use of proprietary InstaCal software libraries. This software package allows users to write their own program via an API (Application Programming Interface) or simply via MATLAB when the correct packages are installed. The board interfaced with the host PC via MATLAB's Data Acquisition Toolbox..

The DAQ board's ADC and DAC can be fed with external, internal or a USB clock. Unfortunately there was one major downside to using this board, that this board is not designed for coherent signal acquisition applications (e.g. coherent radar, coherent sonar) but more for sensory applications (e.g. reading sensors at low sample rates such as accelerometers, thermocouples etc.). Tests showed that, there is an arbitrary phase shift between each channel (input and output, all together). And this phase shift is randomised at every activation of the board. In order to overcome this problem, a reference target was used during the experiments and used to calibrate/align the phases of the signals.

As far as the power requirements go, the board can easily be powered via a 5V DC supply, or simply the USB link.

Also in order to increase reliability, an external clock source was programmed with the use of an Arduino and fed into the clock inputs of the ADC and DAC subsystems. Although this led to an increase in coherency, it did not solve the problem completely, and calibration with real targets was still essential.

4.4.2 Transmitter and Receiver Modules

Transmitter and receiver module designs were made. To maintain compatibility with the DAQ board a 5V supply voltage was selected. To keep the design simple a transformer based amplification circuit was used for transmitter. And for the receive amplification a double-stage low-noise op-amp circuitry with bandpass filtering has been designed. Components were chosen from easy to obtain off-the-shelf components. The schematics of the circuits, the simulations for the transmitter and receiver circuits, and measured receive amplification levels for various frequencies can be found in Appendix B. Note that the appended document was written as a planning document during the preparations of experiments and appended without touching up to ensure the integrity of the document.

The PCB design for the transmitter and receiver boards were also done with modularity and upgradability in mind, therefore slightly more space than necessary was given for the components, and same form factor was used both for transmitter and receiver modules. The blueprints and photos of the final PCB designs along with electronic circuit simulations of the modules can be seen in Appendix B under the title “Acoustic MIMO Technology Demonstrator Implementation”. The phase responses of the modules do not matter as long as they are identical to preserve the coherency among transmitted and received signals. Given that the off-the-shelf components were used and given the wavelength that’s used any production differences in modules have been assumed to be negligible. Photos of the transmitter and receiver PCBs can be seen in Figure 4-3. Finally, as a target, same reference targets that were used in previous chapters were used due to simplicity and availability. Only concern in choosing targets were to make sure they are bigger than the wavelength but not by a massive margin (e.g. larger than the range resolution).

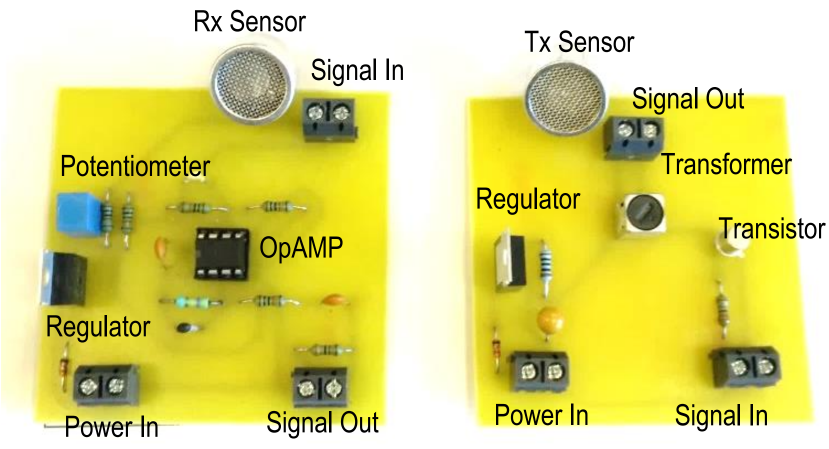


Figure 4-3 Photos of the transmitter (right) and receiver (left) modules

4.5 DESIGN & DEVELOPMENT OF SIGNAL PROCESSING

Most hardware and software blocks were chosen to minimise the coding task required. This includes the use of MATLAB, use of off-the-shelf data acquisition board with open APIs (Application Programming Interface), and use of an Arduino board. In order to test the implemented blocks, the simulator programme from previous chapter was modified to speed up the testing phases and also in order to provide a comparison basis for experiments.

Below is the ordered list of software blocks. Note that the separation between 4th and 5th steps allows the separation of measurement and processing systems, so that the same measurements can be processed differently if needed (i.e. program can be started from step 5).

1. Parameter setup (waveform, array geometry, filters, tapers and windows etc.)
2. Output waveform construction
3. Data acquisition board setup and configuration
4. Firing the outputs and saving inputs to a measurement file
5. Measurement file reading (parameter and signal extraction)
6. Time corrections on signals based on direct path transmission
7. Matched filtering
8. Phase calibration with respect to a known target
9. Beamforming

10. Visualisation (plotting)

There are also processing blocks implemented to overcome problems that are caused by hardware or others. These will be explained in the next chapters with the rest of the implementation details.

4.5.1 Data Acquisition Board Interface

The data acquisition board used is compatible with MATLAB's Data Acquisition Toolbox which sped up the interface development process. The toolbox allows data channels to be opened to ADCs and DACs of the board and the analogue data to be read and written with almost zero transformation. Among the most important capabilities of the toolbox are; setting the sampling frequency, triggering the inputs and/or outputs, and retrieving trigger times in seconds. Ideally the triggering would occur at the same time for inputs and outputs, however initial tests have shown that this is not the case. This happens because of the non-real-time software running on the PC side, causing the inconsistency in timing. Although this problem caused incoherency between the channels, the problem was overcome by using the direct path transmission to align the received pulses in time.

Data Input

When triggered, analogue inputs can capture and save data for a predefined amount of time. The incoming signals were sampled at the carrier level, simply because it would decrease the analogue circuitry requirements (i.e. downconverter, I&Q sampler, etc.). Since it was decided to have 4 input channels and ADC has a 1000 kilosample/second sampling rate shared between these 4 input channels a 250 kilosample/second input sampling rate was obtained per channel. In order to keep parameters coherent, the same sampling frequency was also used for DACs.

The ADCs can operate with 16 bit resolution able to measure voltages between +/- 10 volts or between different ranges down to +/- 100 mV. Since the analogue circuit supplies operate with 0 and 5 volts and the amplifier output is also between those values, it was decided to set the input range between 0 and 5 volts and spread 16 bit resolution to this range, taking advantage of the full capabilities of the input sampling hardware.

As mentioned before, ideally the ADC and DAC clocks would run with the same local oscillator to keep the inputs coherent. However, there is no documentation explaining the internal clocking structure of the board. Initial tests and part of the documentation showed that the ADCs and DACs not only use different clock sources but even the different channels of ADC are incoherent with respect to each other due to the progressive scanning technique (i.e. ADC channels are scanned one after another, therefore there is always a time difference between them) [62].

Data Output

It was previously decided to use 250 kilosample/second input sampling rate and use the sampling frequency for output sampling as well to keep parameters consistent. The DACs can also operate with 16 bit resolution able to synthesise signals between +/- 10 volts. Unlike the ADC this range is fixed, it was made sure that the data vectors are digitally scaled to prevent overdriving the transmitter amplifier circuits.

As mentioned before, there is a coherency problem exists between the channels of ADC and DAC. This problem presented itself again when we tried to separately trigger the output channels to have a coherent sum of the pulses sent from a single channel. In order to fix this problem, it was decided to use a long data vector consisting of all the pulses to be transmitted from a single channel and stream it to the output. This required more memory to be consumed by both MATLAB and the DAQ board but it has proven to be a viable solution.

4.5.2 **Signal Processing of Received Signals**

Signal processing was handled in blocks. This was done to ensure all parts of signal processing was implemented progressively, and that they could be modified/updated without needing to modify everything. The way it works is that each block of signal processing prepares the data for the next step. These steps can be summarised as below:

1. Pre-processing of data and down-converting (i.e. bandwidth filtering, removal of DC signal)
2. Matched filtering to compress the received signals

3. Cropping of all frames of all channels from their first peak to compensate for the major timing differences
4. Application of time differences considering the array element positions (compensation for minor timing differences)
5. Application of the phase shifts computed from expected phases at the range and scan angle of the reference target to compensate for the phase incoherency caused by hardware
6. Application of a beamformer to obtain a range-azimuth map

Pre-processing of Received Signals

Before any radar/sonar processing can be done some pre-processing had to be done to get the signals ready for beamforming.

The first step is the removal of DC signal by subtracting the mean of the received signals. Then all the received signals are filtered with a band-pass filter with a bandwidth that is the same with transmitted signal (4 kHz) centred at the carrier frequency of 40 kHz. Next step is the down-converting to baseband by multiplying the signal with complex sinusoidal with the centre frequency of transmitted waveform. After the down-converting the signal is passed through a low-pass filter. At the end of this block system still has N signal vectors where N is the number of receivers. This summarises the pre-filtering block.

Next block is the matched filtering of the received signals with the transmitted signal(s) to extract the MIMO virtual element signals. This is done via taking the FFT of the received signal and multiplying it with the Fourier transform of the reversed conjugate baseband transmit signal(s), and taking the inverse FFT of the multiplication. This process yields MxNxK many signal vectors where M is the number of transmitters, N is the number of receivers, and K is the number of frames (i.e. number of transmitted pulses for coherent summation). This summarises matched filtering block.

After the extraction of MIMO virtual element signals, first major peak in the matched filter outputs of all frames is found and any data before that point in all vectors is discarded. The reason for that is that the first major peak is assumed to be the direct path signal from the transmitters to the receivers. Logically no reflection can arrive before that time, and due to the close positioning of the array elements this also marks the approximate time of transmission. After the signals are cropped as described, the frames belonging to each virtual element signals are coherently summed to increase the signal to noise ratio. This process yields MxN many signal vectors where M is the number of transmitters, and N is the number of receivers. This summarises the time correction block.

Next step involves the computation of the expected time differences between the direct path signals using the actual physical positioning of the array elements. This time differences are then applied to virtual element signals to compensate for the slightly inaccurate assumption which was made in the previous block. The assumption was that all transmitted signals reach the receivers immediately. This summarises the separation alignment block.

Next step, -probably the most important block of the pre-processing- should not be required with coherent hardware. This block compensates the virtual element signal phases by using a reference target with known range and angle. Knowing the location of the reference target, we can compute the expected dominant signal phases at a specific range assuming the SNR of the target is high enough. Computing the difference between the expected phases and the measured phases at a single point, we then apply these same phase differences to all time vectors to correct the phases of the whole system. This summarises the phase correction block. At the end of this processing block signals are now ready for digital beamforming.

MIMO Beamforming

Multiple beamforming algorithms has been developed for both comparison and verification purposes. The first algorithm that was developed was a modification of conventional sum beamformer for MIMO with added nearfield processing which was already used and verified in previous chapters. Second algorithm is a modification of the Backprojection algorithm for MIMO arrays[63, p. 212]. Backprojection was only used to verify the first algorithm in this thesis and no results obtained with it are presented. Backprojection is usually a reliable SAR algorithm that matched filters the expected phases from a target area to the received signal phases. Usually this means quite a number of matched filters, therefore Backprojection tends to work relatively slow.

4.6 EXPERIMENTAL RESULTS

Upon building a technology demonstrator, a set of experiments were planned and conducted in the anechoic chamber to verify the technology and the demonstrator's capabilities. The experiments were performed in an RF anechoic chamber which still operates reasonably with ultrasonic waves the size of 7 meter long 5 meter wide. The transmitters and receivers were placed on a 65 cm elevation and the targets were placed around the same elevation to avoid 3 dimensional phase shifts.

For sensor placement, a housing with equally spaced slots to hold the receiver and transmitter modules was used. When the sensor modules were placed, the rest of the system was contained in a single box. This kept the technology demonstrator still modular and portable. A photo of the system with the data acquisition box can be seen in Figure 4-4.

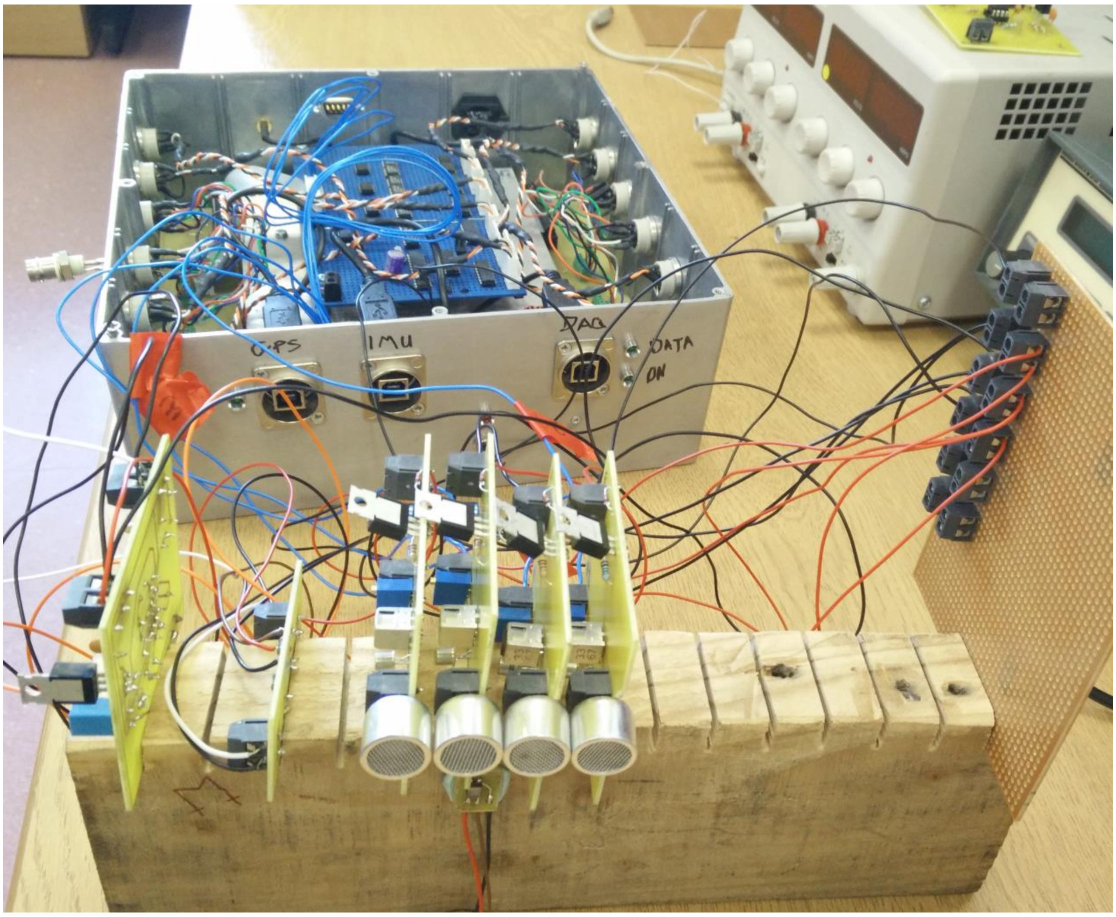


Figure 4-4 Photo of the MIMO sensor array demonstrator with the sensor modules and the data acquisition box

The photos of the experimental setups can be found in the next section demonstrating the results. Below is Table 4-2 summarising the shared parameters of the experiments.

Table 4-2 Experiment parameters

Experiment Property	Value	Unit
Number of Tx	4	-
Number of Rx	4	-
Carrier Medium	Acoustic	-
Carrier Frequency	40	kHz
Wavelength	8.375	mm
Waveform	Up-chirp LFM	-
Bandwidth	4	kHz
Range Window	Gaussian	-
Windowed Range Resolution	0.15	m
Angular Window	None	-
Tx Sensor Beamwidth	55	Degrees
Tx Sensor Bandwidth	2	kHz
Rx Sensor Beamwidth	55	Degrees
Rx Sensor Bandwidth	2	kHz
Receive Gain	~20	dB
Tx Element Spacing	16	mm
Rx Element Spacing	4	mm
Fraunhofer Range	>2.5	m
Sample Rate	250	kilosamples/s
Pulse length	0.08	sec
PRI	0.25	sec
Number of Pulses	3	-

Experimental setup was not ideal. Some of the setup’s shortcomings could be summarised as;

- non-coherent sampling of the analogue data due to acquisition equipment’s characteristics,

- imperfect anechoic chamber,
- reflections from other background elements (equipment itself, chamber door etc.).

Out of these problems the most critical was the non-coherent sampling of the analogue data due to acquisition equipment's characteristics. But it was resolved with aforementioned methods, a demonstration of phase calibration is given in the next section.

4.6.1 Phase Calibration with respect to a Calibrated Target

Virtual element signal phases are compensated by using a reference target with known range and angle. Knowing the location of the reference target, the expected dominant signal phases at a specific range can be computed assuming the SNR of the target is high enough. Computing the difference between the expected phases and the measured phases at a single point, one can then apply the same phase differences to all time vectors to correct the phases of the whole system.

The phases that are introduced to transmitted signals and to received signals by the data acquisition board (i.e. phase mismatch) can be modelled as a constant in addition to virtual element signals as in 4-1.

$$s_{mn_{uncalibrated}}(t) = s_{mf}(t) \cdot p_{mn} \cdot e^{-jkr_m}e^{-jkr_n} \qquad \text{4-1}$$

Where p_{mn} is the complex constant that contains the phase introduced to specific transmitter-receiver channel pair of the data acquisition board. Then, introducing conjugates of these phases would solve the problem as in 4-2.

$$s_{mn_{calibrated}}(t) = s_{mf}(t) \cdot c_{mn} \cdot p_{mn} \cdot e^{-jkr_m}e^{-jkr_n}\,, \qquad c_{mn}p_{mn} = 1 \qquad \text{4-2}$$

Where c_{mn} would be the new phases that'd have to be introduced to matched filtered signals in the datapath to negate the effects of phases introduced by the board. This means there would be M times N many constants to solve for. If the time-of-flight corresponding to reference target's range were to be used as t and if the ideal (expected) signal phases can be estimated at that range using the previously derived models, and using the measured signal phases, then the system can be written as a simple equation as in 4-3.

$$p_{mn}e^{-jkr_n} \cdot e^{-jkr_m} \cdot c_{mn} = e^{-jkr_n} \cdot e^{-jkr_m} \quad \text{4-3}$$

Where the first term $p_{mn}e^{-jkr_n} \cdot e^{-jkr_m}$ is the measured signal phase. So as a final step the constants could be calculated as in.

$$c_{mn} = \frac{e^{-jkr_n} \cdot e^{-jkr_m}}{s_{mn}(t)} \quad \text{4-4}$$

Where $s_{mn}(t)$ is the measured signal phase at the target range. Then, it becomes then trivial to compute each c_{mn}. And once these constants are applied to entirety of the received signals, then theoretically the equipment would be calibrated.

In Figure 4-5 an example setup can be seen. The expected phase differences can be computed from a point target residing at 1 meters, so necessary phase shifts can be computed to make the return to resemble a Sinc function. Then applying these phase shifts to incoming signals give us what we expect at 1 meters range, and inherently all other range samples.

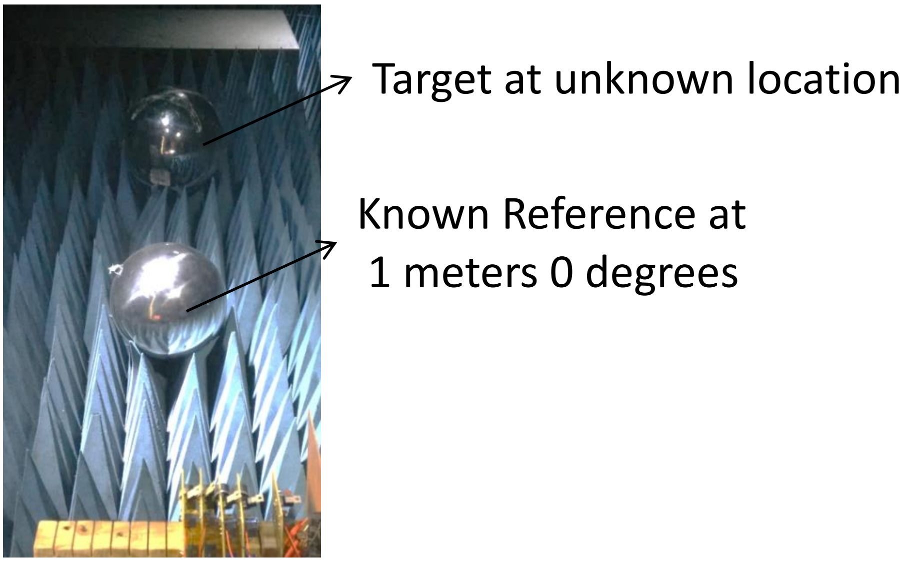


Figure 4-5 Example setup of placing a phase calibration target in experiment chamber

In Figure 4-6 the range-azimuth map obtained via beamforming without calibration can be seen. Even though there are strong reflections at the ranges one would expect, the beam pattern does not form. This is because the received signals are not phase coherent. But taking a single range sample from all virtual element signals and then calculating the difference from expected (computed) phase differences the virtual elements yields the necessary phase shifts to bring back phase coherency between signals.

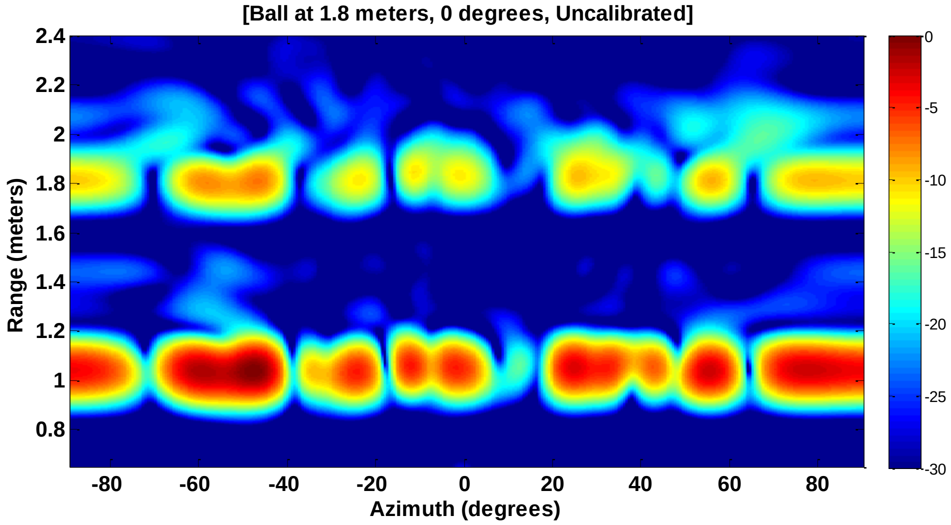


Figure 4-6 Range-azimuth map obtained via uncalibrated use of incoming signals

In Figure 4-7 the range-azimuth map obtained via beamforming with calibration can be seen. In this range-azimuth map both the reference target and the target with unknown location can be identified and resolved properly. Applying these phase shifts to entire range vectors to all virtual element signals and then beamforming as if the signals were coherent yields near-ideal results.

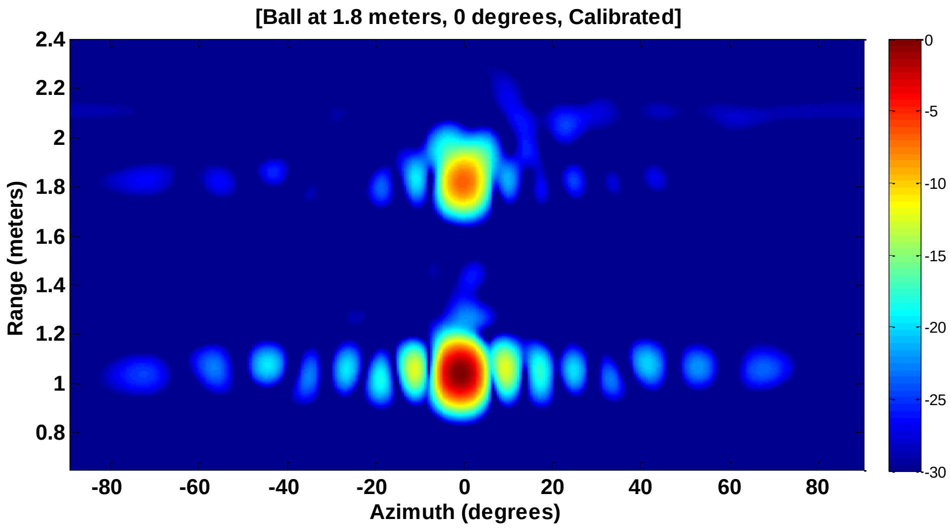


Figure 4-7 Range-azimuth map obtained via calibrated use of incoming signals using the reference target

This method has been applied to deal with phase incoherency caused by hardware in this chapter and the next one. For the sake of simplicity and to reduce the dynamic range (to be able to focus on actual target measurements), reference targets have been cropped out from the results.

4.6.2 **Scenarios and Results**

Various experiments were performed where target(s) were placed about 1.8m away from the sonar and changing azimuth angles. Below are the 5 scenarios simulated with various configurations and experimented with a 4x4 MIMO sonar. The scenarios are presented with from-the-top sketches and photos taken inside the anechoic chamber during the experiments. Next to the top view sketches of setups are the photos taken inside the anechoic chamber during experiments. These scenarios are designed to check beamforming and target localization capabilities, azimuth resolution and finally multiple target resolution capabilities. After each scenario presentation, the results obtained with near-field MIMO processing are presented.

Single Target Experiments

These scenarios were designed to verify single target detection capabilities at different scan angles (beamwidth, sidelobe levels, etc.), and to verify targets are observed on the right direction (i.e. left/right).

Single Target 0 degrees

In Figure 4-8 a top-view sketch of the first scenario can be seen. A target is placed at about 1.8m slant range with about 0 degrees angle. The photo of the setup can be seen in Figure 4-9.

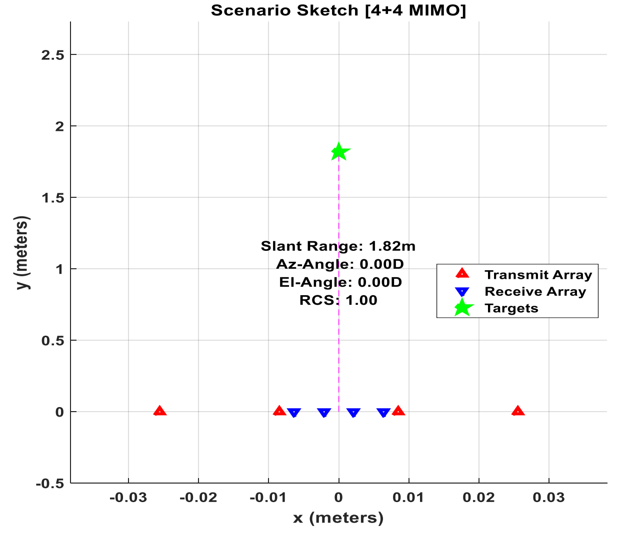


Figure 4-8 Single target at 0 degrees; scenario sketch in Cartesian coordinate system

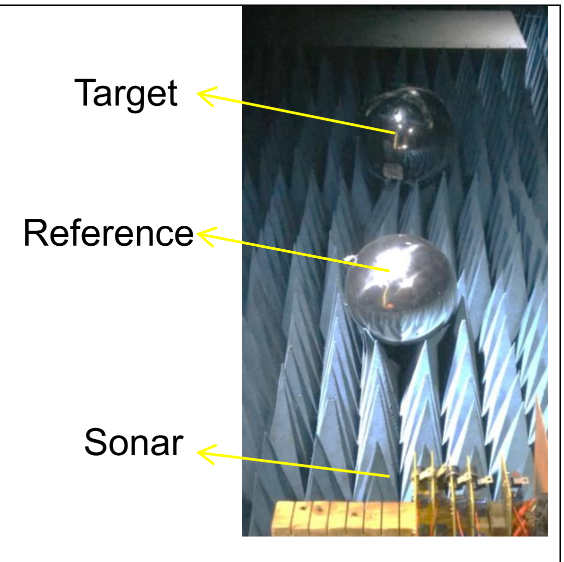


Figure 4-9 Single target at 0 degrees; photo taken from sonar's point of view

Below in Figure 4-10 is the range-azimuth map obtained from measurements with range-gating the phase calibration reference and without any post-processing method. In the figure, a strong reflection at 0 degrees and 1.8 meters can be observed. This location coincides with the target's actual location.

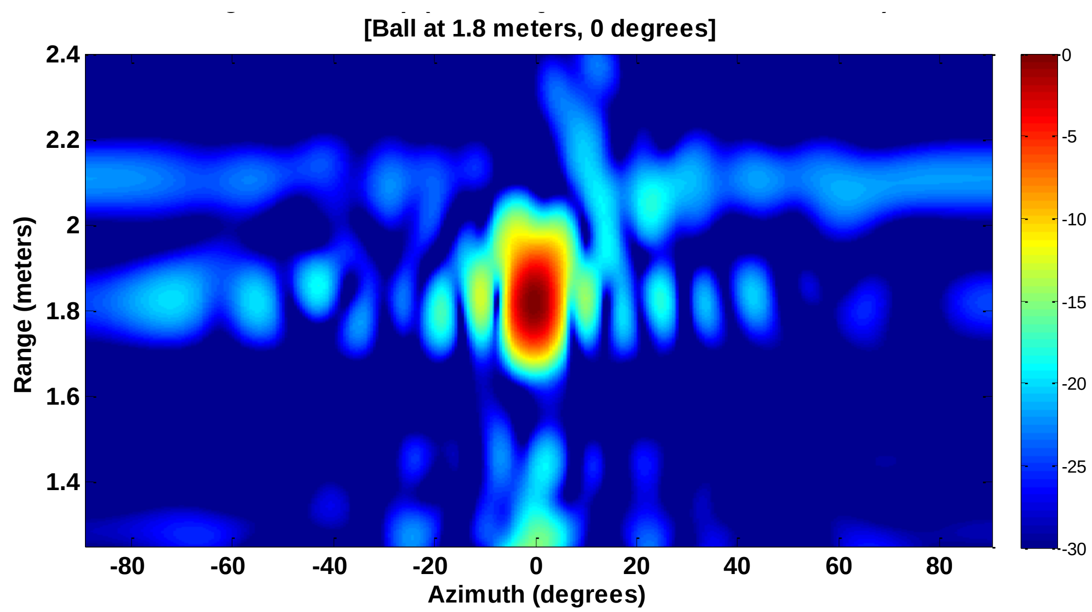


Figure 4-10 Single target at 0 degrees; range-angle map

In Figure 4-11 the azimuth cut from the range-angle map at the target range of 1.82 meters can be seen. The beamwidth is measured is about 6.5 degrees, which coincides with the expected beamwidth of 6.7 degrees. Also in the angle cut the sidelobe level can be seen as -12.8dB which is in 10% vicinity of the usual -13.1dB sidelobe levels.

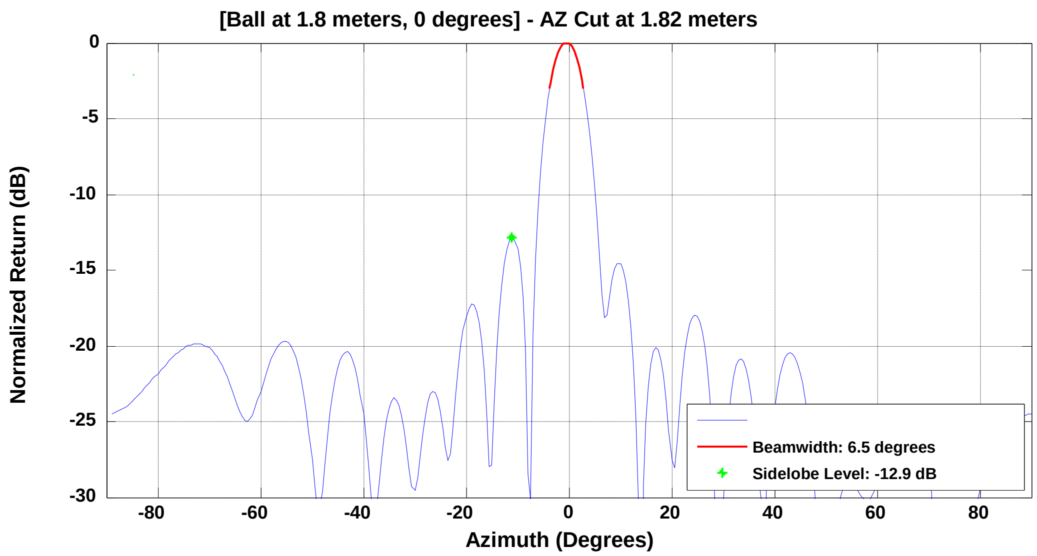


Figure 4-11 Single target at 0 degrees; azimuth cut

<u>Single Target at 15 degrees</u>

In Figure 4-12 the over the top sketch of the second scenario can be seen. A target is placed at about 1.8m slant range with about 15 degrees angle. The photo of the setup can be seen in Figure 4-13.

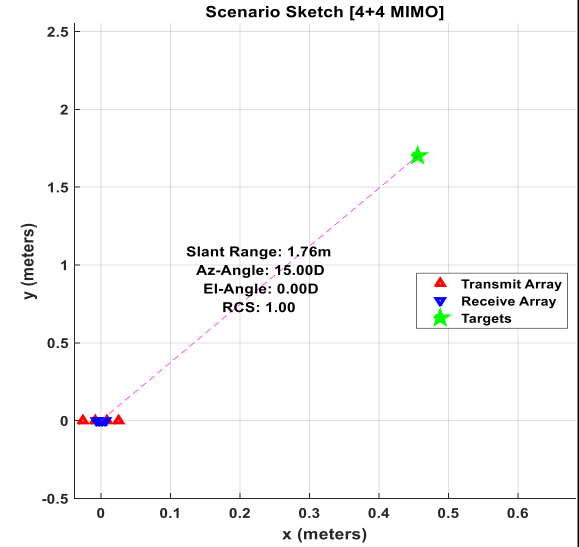


Figure 4-12 Single target at 15 degrees; scenario sketch in Cartesian coordinate system

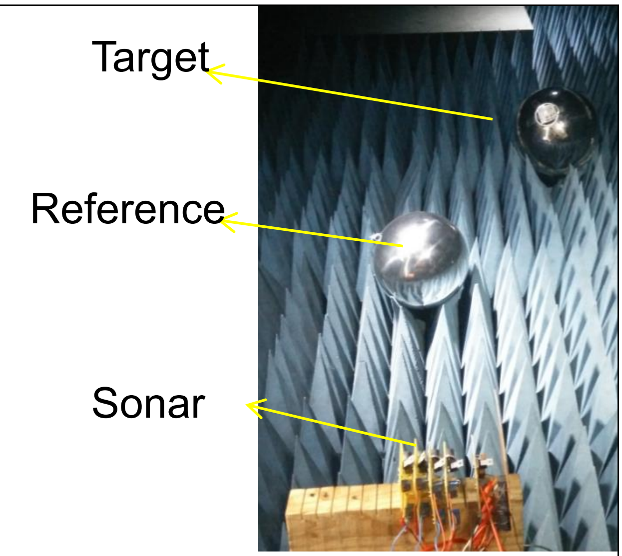


Figure 4-13 Single target at 15 degrees; photo taken from sonar's point of view

Below in Figure 4-14 is the range-azimuth map obtained from measurements with range-gating the phase calibration reference and without any post-processing method. In the figure, a strong reflection at about 15 degrees and 1.8 meters can be observed. This location coincides with the target's actual location.

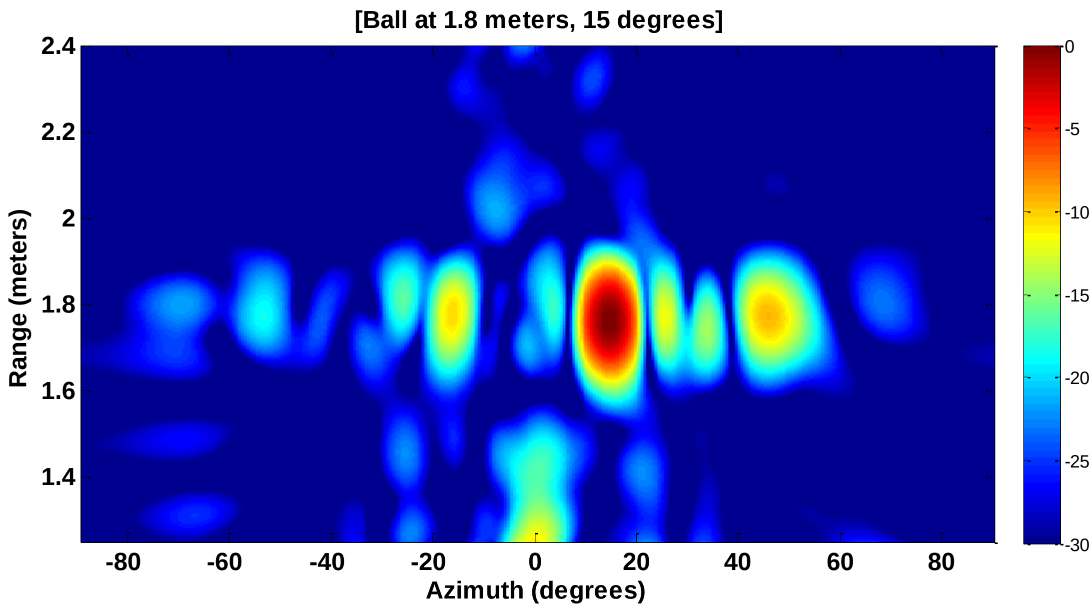


Figure 4-14 Single target at 15 degrees; range-angle map

In Figure 4-15 the azimuth cut from the range-angle map at the target range of 1.76 meters can be seen. The beamwidth is measured is about 6.7 degrees, which coincides with the expected beamwidth of 6.9 degrees at this scan angle. Also in the angle cut the sidelobe level can be seen as -9.4dB which is more than expected, however this can be explained due to the imperfections in the sensor positioning, non-ideality of anechoic chamber and reflections from the reference.

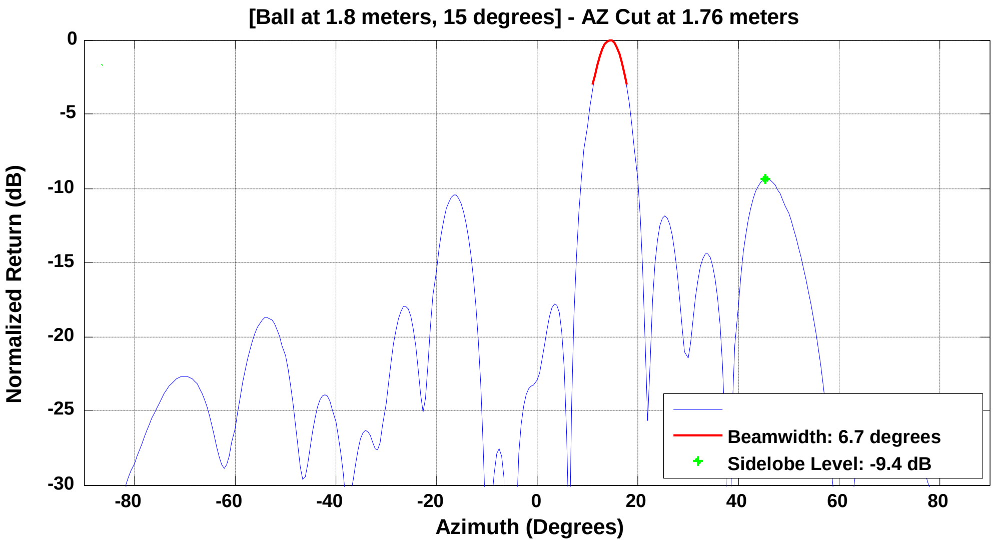


Figure 4-15 Single target at 15 degrees; azimuth cut

<u>*Single Target at 25 degrees*</u>

In Figure 4-16 the over the top sketch of the third scenario can be seen. A target is placed at about 1.9m slant range with about 25 degrees angle. The photo of the setup can be seen in Figure 4-17.

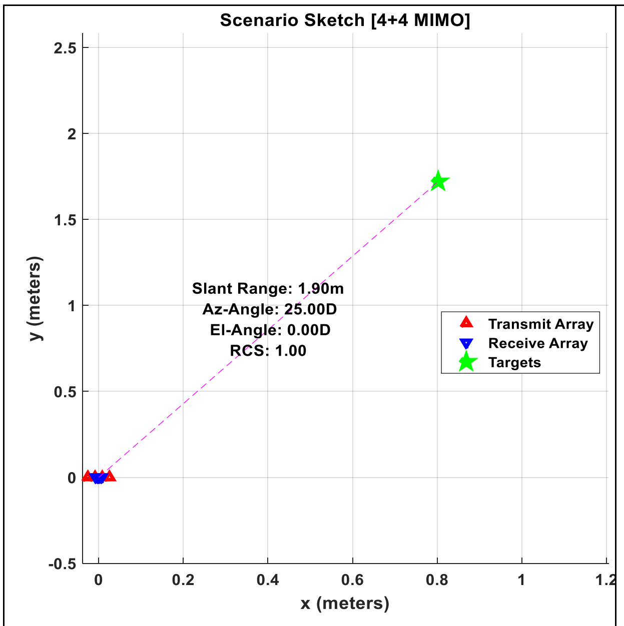


Figure 4-16 Single target at 25 degrees; scenario sketch in Cartesian coordinate system

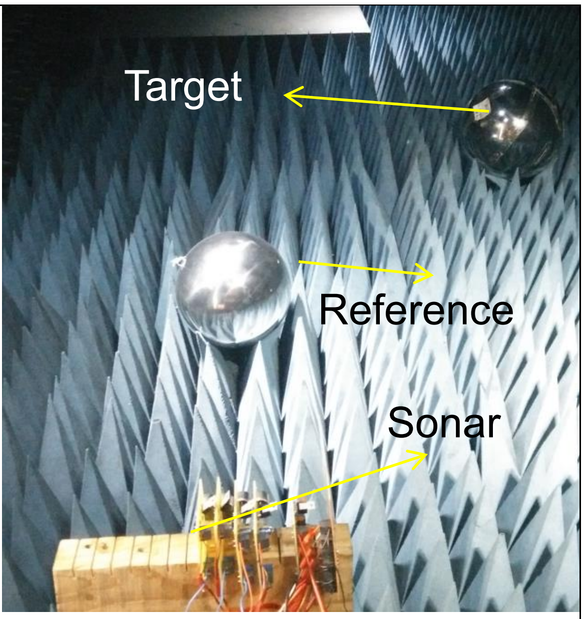


Figure 4-17 Single target at 25 degrees; photo taken from sonar's point of view

Below in Figure 4-18 is the range-azimuth map obtained from measurements with range-gating the phase calibration reference and without any post-processing method. In the figure, a strong reflection at about 23 degrees and 1.9 meters can be observed. This location coincides with the target's actual location.

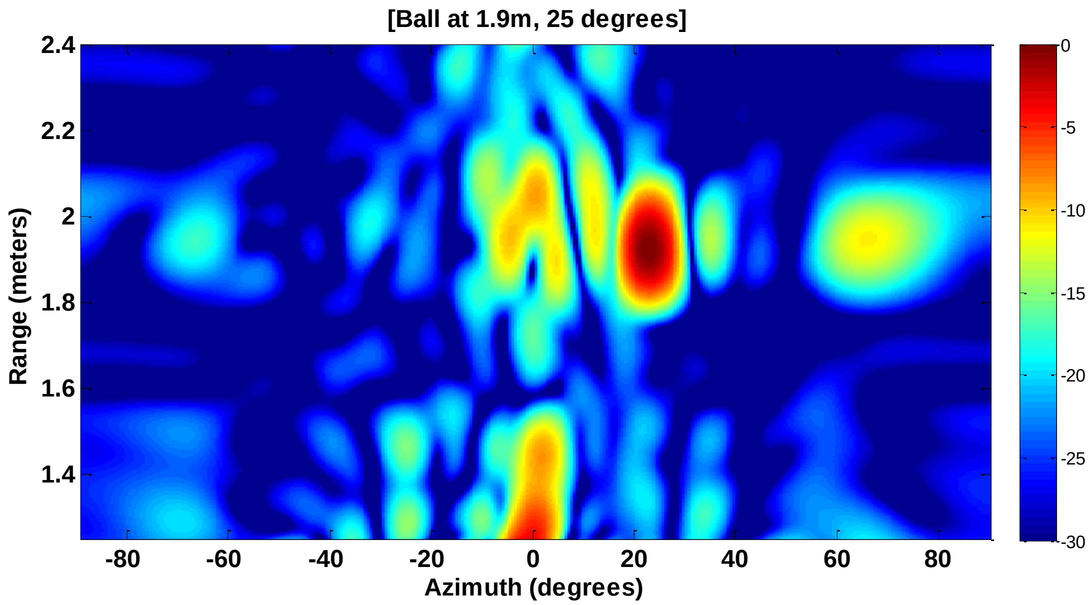


Figure 4-18 Single target at 25 degrees; range-angle map

In Figure 4-19 the azimuth cut from the range-angle map at the target range of 1.76 meters can be seen. In the angle cut beamwidth is measured is about 6.9 degrees, which coincides with the expected beamwidth of 7.4 degrees at this scan angle. Also in the angle cut the sidelobe level can be seen as -

9.9dB which is more than expected, however this again can be explained with the imperfections in the sensor positioning, non-ideality of anechoic chamber and reflections from the reference (i.e. multi-path interference, range and angle sidelobes adding up etc.).

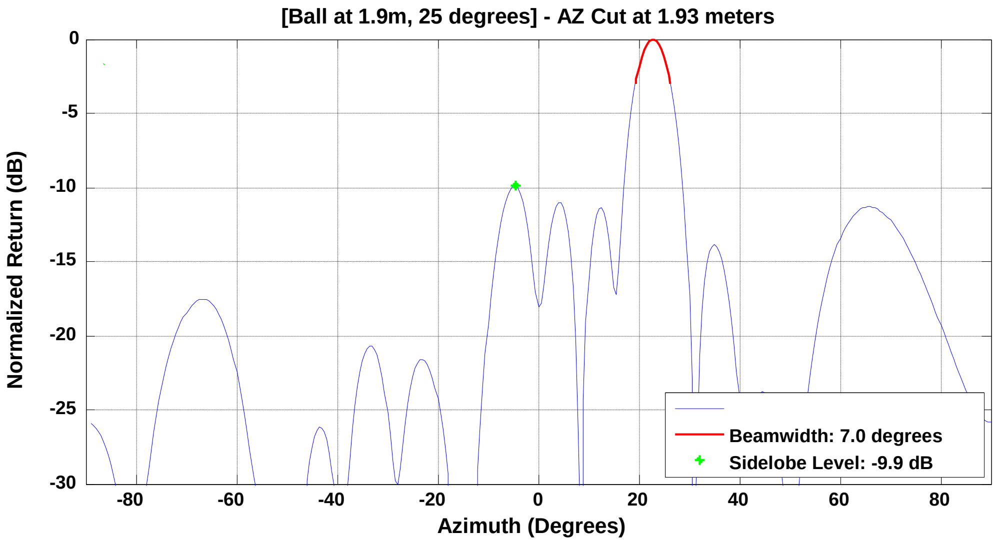


Figure 4-19 Single target at 25 degrees; azimuth cut

Considering all single target experiments that have been performed, an increase in the beamwidth was observed as the target move further away from the zero angle, which is expected in scanning sensor arrays. Also in comparison we start to observe that background artefacts become stronger as target moves further away from zero angle. This doesn't mean that the background is getting stronger, however the return from target is getting weaker since range-angle maps are normalised with respect to the strongest reflection within. We assume this is due to the non-isotropic beam pattern of the transducer that's used.

Multiple Target Experiments

These scenarios was designed to verify multiple target resolution capabilities and to observe how they interact. Two targets were mainly placed in close proximity in angle and range.

Two Targets at 0 and 20 degrees

In Figure 4-20 the over the top sketch of the fourth scenario can be seen. Two targets were placed at about 1.8 meters and 1.6 meters slant range with about +1.5 and +21.6 degrees angles respectively. The photo of the setup can be seen in Figure 4-21.

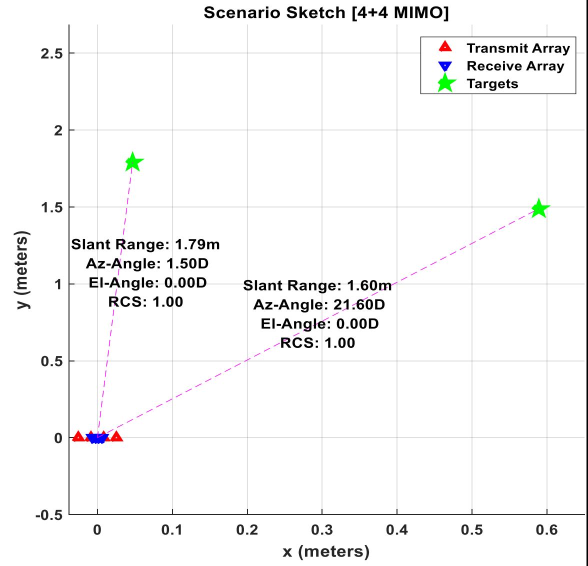


Figure 4-20 Two targets at 0 & 20 degrees; scenario sketch in Cartesian coordinate system

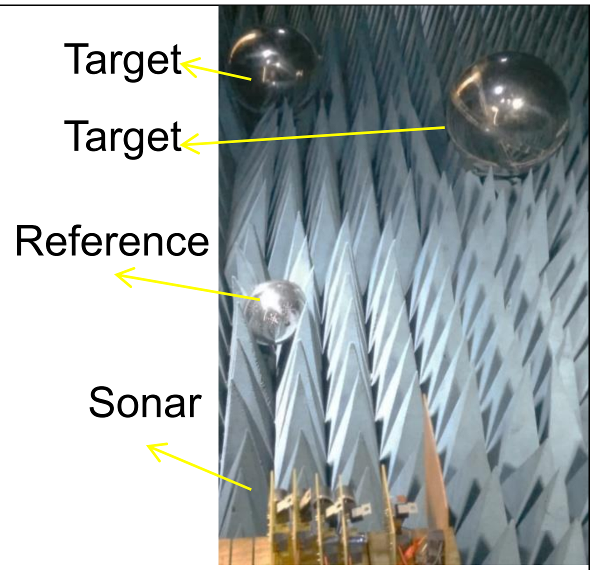


Figure 4-21 Two targets at 0 & 20 degrees; photo taken from sonar's point of view

Below in Figure 4-22 is the range-azimuth map obtained from measurements with range-gating the phase calibration reference and without any post-processing method. In the figure, 2 strong reflections can be observed, highest at 0 degrees, 1.8 meters and second highest at 20 degrees, 1.6 meters. These locations coincide with the targets' actual locations.

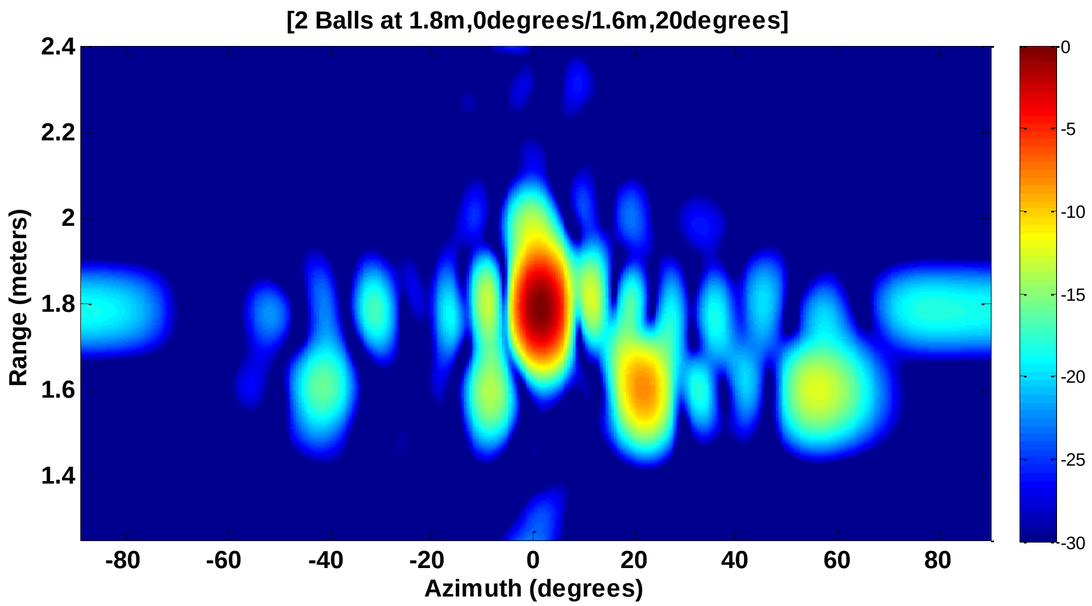


Figure 4-22 Two targets at different ranges; range-angle map

In Figure 4-23 the azimuth cut from the range-angle map at the slant range of 1.6 meters - which is the target range for the strongest reflection- can be seen. The beamwidth for 0 degree scan angle is measured about 6.4 degrees which is in 10% vicinity of the expected beamwidth of 6.7 degrees. Also in the azimuth cut the sidelobe level for the target at 0 degrees can be seen as -12.9 dB.

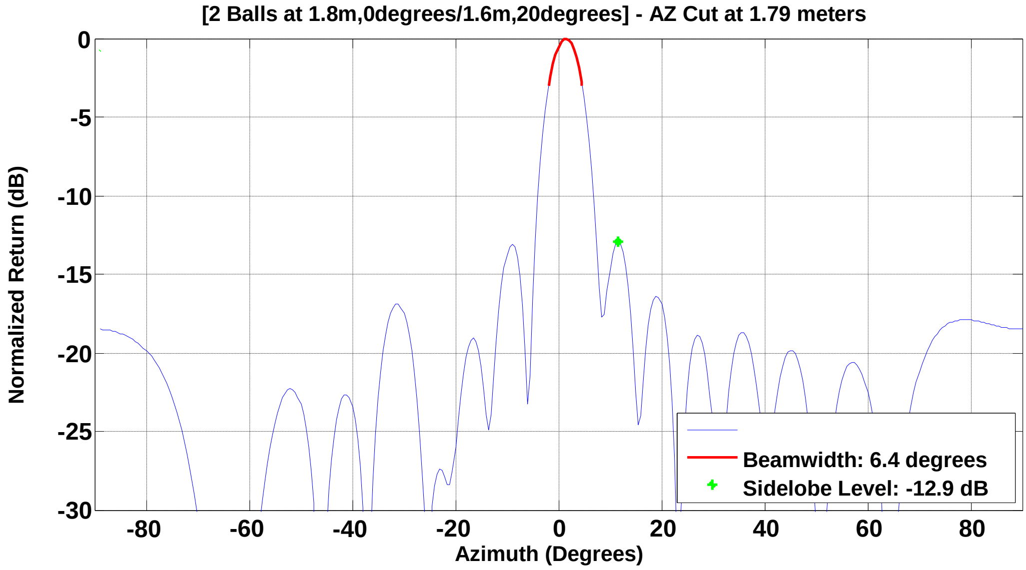


Figure 4-23 Two targets at different range; azimuth cut at 1.6 meters

In Figure 4-24 the azimuth cut from the range-angle map at the target range of 1.79 meters can be seen. The beamwidth for 20 degrees scan angle is measured about 7 degrees which is quite close to the expected beamwidth of 7.1 degrees at this scan angle. Also in this azimuth cut, grating lobes of the coarse can be observed at the levels of -4.4dB; this is assumed to happen due to a mismatch with fine array's nulls.

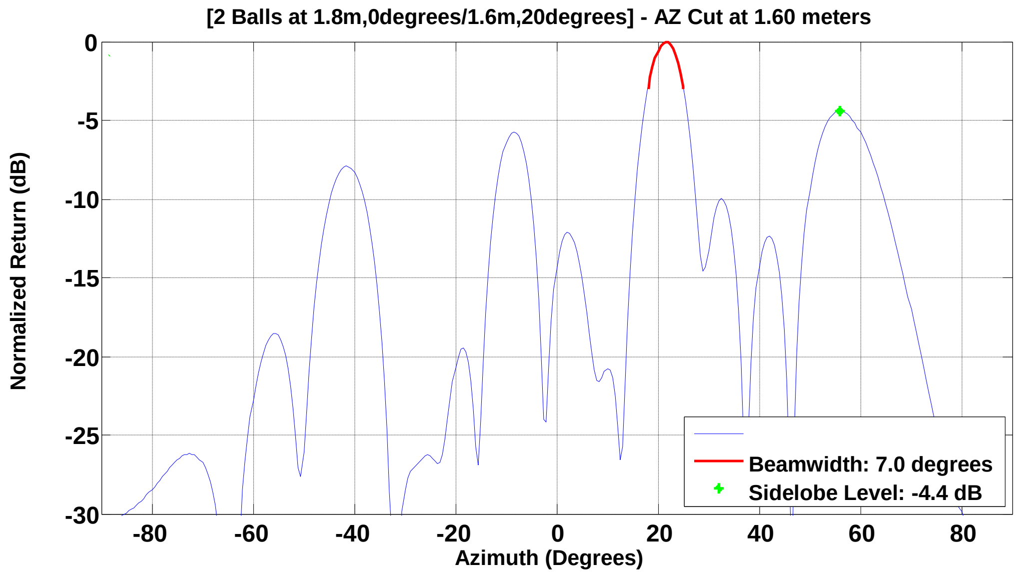


Figure 4-24 Two targets at different range; azimuth cut at 1.8 meters

Two Targets at 0 and 25 degrees

In Figure 4-25 the over the top sketch of the fourth scenario can be seen. Two targets were placed at about 1.8 meters slant range with about 0 and +25 degrees angles respectively. The photo of the setup can be seen in Figure 4-26.

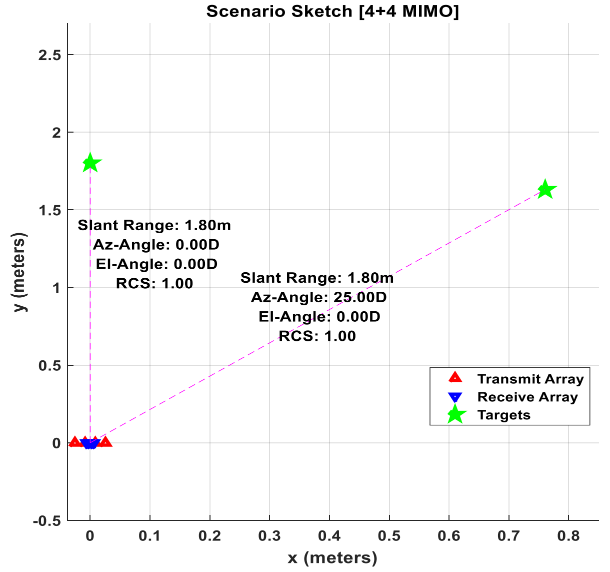


Figure 4-25 Two targets at 0 & 25; scenario sketch in Cartesian coordinate system

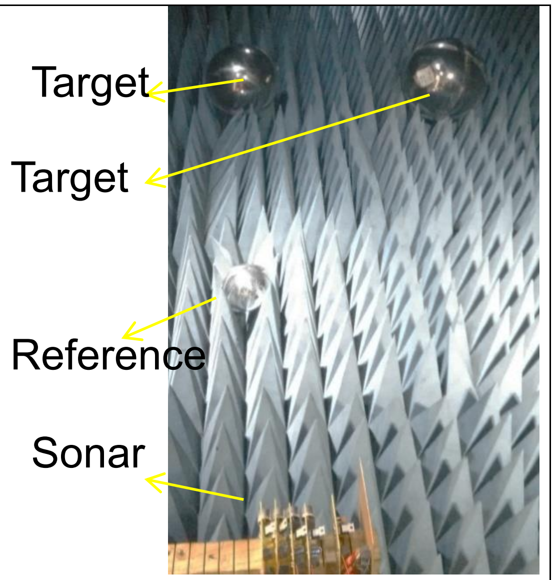


Figure 4-26 Two targets at 0 & 25; photo taken from sonar's point of view

Below in Figure 4-27 is the range-azimuth map obtained from measurements with range-gating the phase calibration reference and without any post-processing method. In the figure, 2 strong reflections can be observed, highest at 0 degrees, 1.8 meters and second highest at 25 degrees, 1.9 meters. These locations coincide with the targets' actual locations.

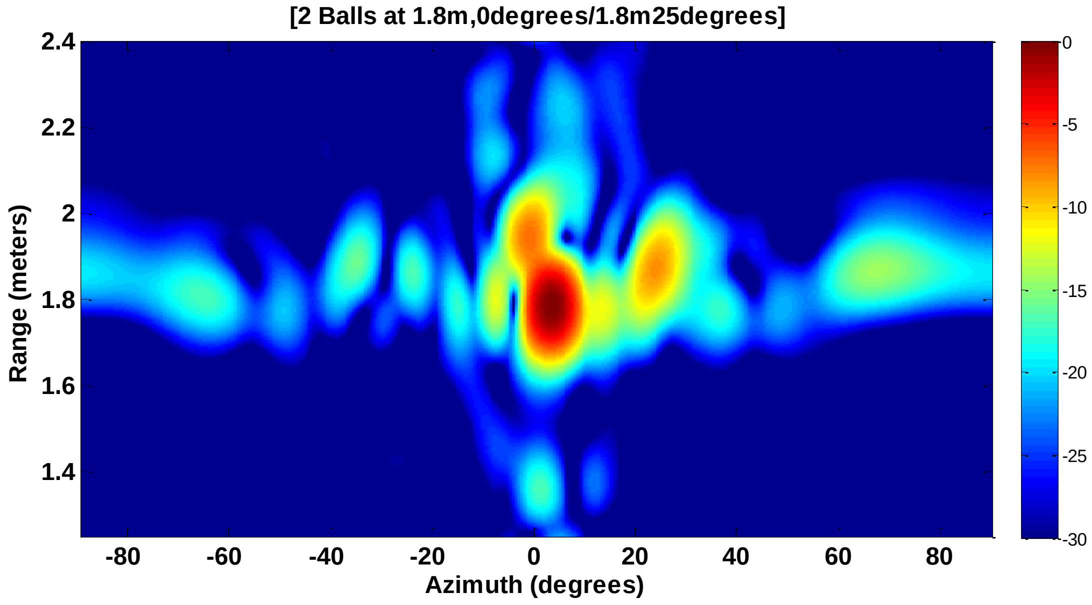


Figure 4-27 Two targets at similar range; range-angle map

In Figure 4-28 the azimuth cut from the range-angle map at the slant range of 1.79 meters - which is the target range for the strongest reflection- can be seen. Note that the azimuth cuts are normalised to their own maxima. The beamwidth for 1.5 degree scan angle is measured about 6.8 degrees which is quite close to the expected beamwidth of 6.7 degrees at this scan angle. In figure the sidelobe level for the target at 0 degrees can be seen as -11.2dB which is more than expected, but only 2 dB less than the usual -13.1 dB sidelobe levels. It can be observed that there is an asymmetry on the right side at 25 degrees. It is assumed this is actually part of the other target's return which is at 25 degrees.

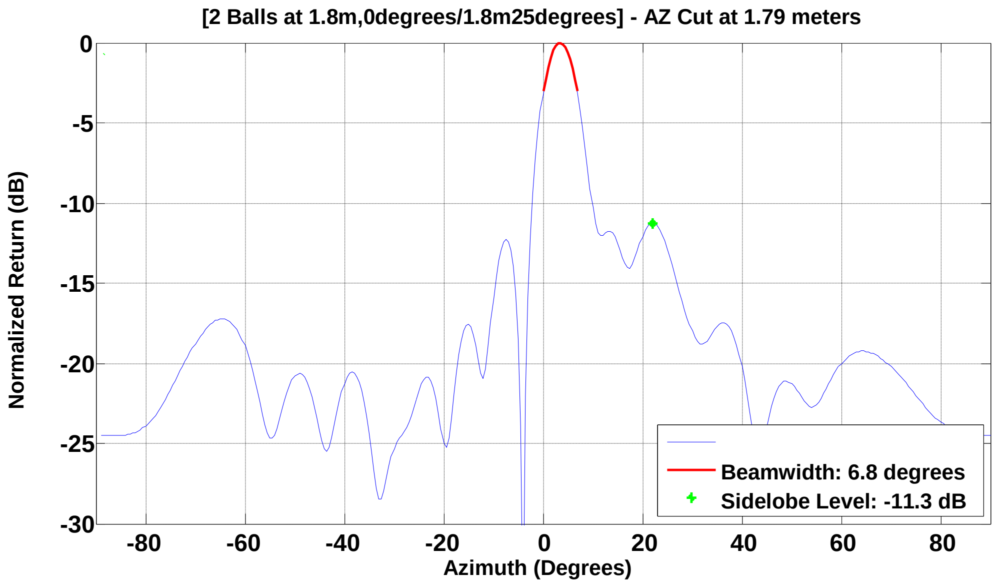


Figure 4-28 Two targets at similar range; azimuth cut at 1.79 meters

The beamwidth for 25 degrees scan angle is measured about 7.8 degrees which is in the 10% vicinity of the expected beamwidth of 7.36 degrees at this scan angle. In the figure the sidelobe level for the target at 25 degrees cannot actually be measured because the main lobe and the first sidelobe of the first target's main signature is quite strong on the angle cut at this range. Such that it's even stronger than this target's main signature.

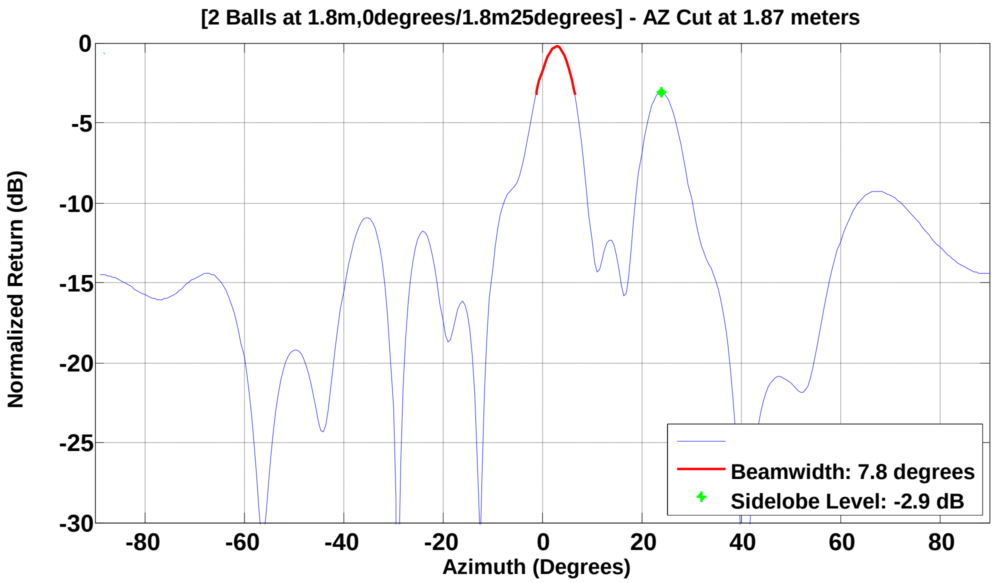


Figure 4-29 Two targets at similar range; azimuth cut at 1.87 meters

Similar to what's observed in single target experiments is that targets further away from zero angle have less reflectivity compared to their zero angle counterparts. Another observation is the sidelobe interaction of multiple targets. Range sidelobes and angle sidelobes do tend to add constructively to create strong reflection points in the range-angle map. In the presence of multiple targets with different radar cross sections, it might not always be easy to spot the actual targets from these returns.

4.7 Conclusions

After testing the MIMO sensing theory with existing hardware, a technology demonstrator design was started with automotive applications in mind. It was decided to have 4 transmit elements and 4 receive elements and to have acoustic sensors instead of RF. An acoustic MIMO sensor array was designed and built as technology demonstrator. This technology demonstrator was built with modularity in mind to allow for upgrades. In order to achieve this goal, transmit and receive modules are designed as separate PCB boards from each other and a separate housing for these boards was built, so that the element spacing could be adjusted as easy as building a new housing.

Along with the hardware design, software design was done simultaneously. A simulation program was implemented and used to develop signal processing algorithms until experimental setup was ready, and used to verify the experimental results after the setup was ready. As the hardware became ready, it was tested and its problems were identified and patched either via software (e.g. timing corrections with respect to direct path signal) or hardware (e.g. use of an Arduino board for clock generation, addition of a reference target during experiments).

Multiple experiments were conducted to verify that MIMO sensing theory is still applicable in ultrasonic domain and also to verify the demonstrator works nominally. The experiments resulted in minimal deviation from expected results and the targets were identifiable in ground scans of anechoic chamber at the expected locations with expected performance metrics (e.g. beamwidth, sidelobe levels).

5 MIMO SENSOR ARRAY OPTIMISATION & RESULTS

5.1 INTRODUCTION

In the previous chapter, a technology demonstrator was built and tested experimentally. Specifically, an acoustic MIMO sensor array was built with modularity in mind so that more optimised array configurations could be explored. The technology demonstrator was designed and implemented to meet potential automotive application requirements.

Next step in research was to explore thinned-like arbitrary MIMO array configurations to obtain further performance optimisations in array sensing. For this purpose, heuristic search algorithms were explored and implemented. Heuristic algorithms work to find a good solution to a given problem without doing an exhaustive search and so they don't scan the entire solution space, which usually saves time but they're not always guaranteed to find a good or the best solution. Several promising array configurations were reached, which were tested via simulations. Finally, a configuration that promised the most optimisations were chosen to be implemented and tested with the same conditions as before to experimentally prove our methods. The experiments resulted in minimal deviations from the expected results.

In this chapter, an introduction to heuristic search algorithms are given and how they can be used to optimize MIMO array geometries are explained. The MIMO array already yields less cost/higher performance when compared to its phased array counterpart (i.e. same physical aperture size). To further optimize the performance of MIMO arrays, it was found useful to consider thinned array configurations. This is usually useful to further minimise the amount of physical elements needed for high resolution sensing, which is desirable in an automotive context. However, since our technology already has minimal amount of physical elements both in transmit sub-array and receive sub-array, a different approach was taken rather than thinning. That is, rather than trying to cut down element numbers, we are trying to find the maximum virtual array length occupied with the number of elements already available. So, optimisation starts already with a low number of elements, which is unlike most similar research.

To achieve this task, various optimization algorithms were tried to find solutions; namely random descent, simulated annealing and genetic algorithm. Our implementations of random descent and simulated annealing have successfully yielded us practical solutions. The results obtained from these algorithms were then fed to our implementation of a genetic algorithm to possibly get further optimisations. Examples of configurations obtained via these algorithms are presented and our conclusions about the optimisation process is explained. Then the configuration with the most optimisation was chosen to be implemented and verified with an experimental setup like in previous chapter. The results obtained from the experiments and conclusions are given at the end of this chapter.

5.2 OPTIMISATION APPROACH

In engineering contexts, a heuristic approach to a problem is a method to solve a problem when conventional and/or analytical methods are too slow or an exact solution is not easily obtainable. A heuristic approach trades off from the quality of the solution for speed. They are essentially optimization algorithms as an alternative to exhaustive search algorithms. And in return they don't always come up with the "best" solution but with "good" solutions which are still acceptable. In contrast to exhaustive search, a heuristic algorithm doesn't try all possible solutions but eliminates some of them based on what it already has tried (or sometimes it simply scans random solutions) [64].

Arbitrary array geometries are often and widely used to produce desirable beam patterns[13, p. 619]. However, computation of array geometry from a desired pattern or specifications (such as null and lobe positions, beamwidth, maximum scan angle etc.) is not always a straight-forward process. For this purposes, various search algorithms are used to go through a number of array geometries to obtain a desirable or close-to-desirable solutions[65][66][67]. Apart from obtaining desirable patterns, various search algorithms are also used to generate patterns for thinned arrays[68][69][20][70][71]. And finally in MIMO array context, heuristic algorithms also are being used for array thinning and obtaining desirable patterns (e.g. particle swarm, cyclic permutation of perfect distance, genetic algorithm)[72][42][45][40].

When the search space is small, it is often feasible to use exhaustive search depending on the available computation power, but when the number of variables increase the complexity of the problem also increases in an exponential order. It becomes infeasible to search through all variations of variables (i.e. all possible array geometries). This is where it becomes necessary in practice to use an advanced heuristic search algorithm to go through the solution space in a more computationally efficient manner.

In this case, we are dealing with MIMO arrays, which means array element positions convolve in space to give a virtual array with a given beam-shape and then generate patterns [73]. This process makes pattern design more complex compared to phased arrays.

The approach starts with basic heuristic algorithms to address the problem. Firstly, a basic heuristic algorithm (e.g. random descent) was implemented to generate patterns for a MIMO array with given constraints and goals (e.g. minimum beamwidth or minimum sidelobe levels). After the verification of the algorithm via simulations, more advanced heuristics were implemented.

Simulated Annealing (SA) algorithm is a rather simple yet effective and relatively simple to implement heuristic algorithm. As a next step SA was implemented to generate patterns for a MIMO array with given constraints. After the fine tuning and verification of the algorithm via simulations, final stage of optimisations was commenced.

As a final implementation a Genetic Algorithm was implemented and fine-tuned to generate patterns for a MIMO array with given constraints. After the fine-tuning and verification of the algorithm via simulations, the outputs were put through simulations and confirmed via experimental testing.

For testing and verification of the algorithms; the optimizers could be asked to solve more trivial problems such as optimizing shorter arrays where exhaustive searchers (or humans) could solve it too. And this could be used to verify optimizers work for small problems. But unfortunately, most optimization algorithms (and the ones proposed here) are not linear algorithms and they don’t necessarily scale up. Still, the best way to test the optimization algorithms, is to run them for the needed problem and to see if they offer any usable solutions and then increase the processing time. All algorithms were tested with the specific problem that we're trying to solve, however for testing and verification purposes they were tested with "easier" search parameters to lower their processing time during development and verification. "Easier" search parameters in this context could mean; lowered solution sets to search through, lowered time limits and lowered number of iterations. These limits are imposed on the algorithms in the form of lowered initial temperature, increased cool-down speeds, lowered number of generations, lowered population sizes, lowered probabilities and similar (the definitions of these parameters will be explained in the next section)

5.3 Algorithm Descriptions

First, the terminology used to explain heuristic optimisation algorithms is explained. But since the scope of this research is not about developing heuristic algorithms, the algorithms and implementations are still explained within MIMO sensor array context. After laying out the definitions and terminology, the algorithms and their exact implementations are given.

5.3.1 Definitions and Keywords

A *solution* is a set of variables stating a configuration of the array. In our specific case, the variables are the positions of the array elements in terms of wavelength. Note that even though a solution is called a solution, it doesn't have to be a feasible one (i.e. does not have to be a working solution). It simply denotes a configuration (of array element positions).

Fitness, cost, score and similar terms are used to define the "*wellness*" of the solution. In our case, if the optimisation goal was to reduce the beamwidth, the fitness function would be related to beamwidth. If the optimisation goal was to minimise the maximum relative sidelobe level (mRSLL), the fitness function would be related to sidelobe levels. Or in more complex configurations, fitness or cost function would be related to number of results including beamwidth and mRSLL, and the goal would be to minimize the cost or to maximize the fitness.

Perturbation, *jump* or an *operation* is a predefined operation on a single or multiple variables of a solution to obtain a new solution; a *neighbour*.

A *neighbour* of a solution is a solution obtained from a solution by making a predefined operation on the original solution. These operations can be -but not limited to- randomizing a variable in the solution, randomizing all the variables in the solution, incrementing or decrementing a variable (array element positions) with a fixed predefined step in the solution etc.

A *population* or a solution *set* -as the name suggests- is simply a set of solutions. All, none or some of these solutions can be feasible.

Current solution is a special solution in which an algorithm makes jumps from. Depending on the algorithm there may be more than a single current solution. In that case it may be referred to as *current solution set*.

A *neighbourhood* is a solution set obtained from an original solution by applying a single predefined operation with different values and/or to different variables (array element positions).

Feasibility of a solution is defined if the solution actually works. The feasibility is not always defined but in our case, it is necessary. Although "black and white" results often confuse search algorithms, a feasibility line was drawn because some solutions yield completely unusable beam patterns. And in order to avoid confusing the algorithms, feasibility of a solution is implemented in a way that it decreases the fitness of the solutions by a high margin. So that an algorithm can still pick the best of the worst if it's traversing through a "bad neighbourhood".

A *local optimum* is the best solution in a given neighbourhood. If an algorithm is not able to make jumps between neighbourhoods, it can get stuck in that neighbourhood. For example, if an algorithm was set to narrow the beamwidth of a MIMO array and if it starts with the conventional MIMO configuration, the algorithm is likely to reach time or iteration limit without finding an improvement, because conventional MIMO configuration is already a "good" solution and possibly the best solution in a neighbourhood of simple jumps (i.e. starting from a conventional MIMO configuration, moving a single array element randomly will almost never result in narrowing the beamwidth).

Acceptance or *acceptation* is the scenario when a neighbour is accepted to be the next current solution or to be included in the next current solution set. This neighbour doesn't have to offer a better fitness value. Acceptance can be completely probabilistic (e.g. random descent), or it can be simply denied based on a set of rules (e.g. Tabu search), or it can simply occur because it offers better (e.g. first descent) or best (e.g. steepest descent) fitness values etc.

5.3.2 Optimisation Goals

It's been found that there are two general approaches to select from and start optimising. First is having a fixed aperture size (and therefore a relatively fixed beamwidth) and optimising (minimising) the sidelobe levels. Second is having a maximum sidelobe level and optimizing (minimising) the beamwidth (and therefore maximising the aperture size). In either case the algorithm's approach is similar but with different constraints. In all iterations some feasibility checks are done to ensure and enforce that solutions are practical. These feasibility checks can be summarized as;

- Nonexistence of grating lobes
- Fulfilling the maximum beamwidth requirement (applicable to variable aperture size)
- Fulfilling the maximum sidelobe level requirement (applicable to fixed aperture size)
- Having a correct beam direction
- Having acceptable edge sidelobe levels (sidelobe levels at maximum scan angles)

These checks are performed for multiple scan angles including zero, negative and positive maximum scan angles. Failure in these checks severely damages the fitness of the solution. If a solution fails in all of these checks in all of the possible scan angles then it gets the worst possible score. If a solution fails only in some of these checks then it rises amongst the infeasible solutions to make path for a better neighbourhood.

Sidelobe Level Minimization with Fixed Aperture Size

To set a fixed aperture size, we first set the transmit array to be the coarse array and make the first and last elements have fixed positions. In such way we enforce fixed physical aperture size to the array. The rest of the element positions would be conventional MIMO array positions. Intuitively this yields a non-optimal solution but ultimately a solution which is not in a "bad" neighbourhood. With this adjustment we also end up with a relatively fixed beamwidth too.

To run this scenario a fixed aperture size and a maximum acceptable beamwidth has to be entered by the user as a parameter.

If there are M transmitter and N receiver elements to be optimized this configuration yields M-2+N variables to play around with. Those variables have a range from zero (position of the first fixed element) to aperture size (position of the second fixed element).

Beamwidth Minimization with Maximum Acceptable Sidelobe Level

Beamwidth minimization is achievable through maximizing the physical aperture. The initial solution would be the conventional MIMO array configuration. Then algorithms try solutions of array configurations with lengths up to a predefined maximum aperture size. A key constraint here is the maximum acceptable sidelobe level. Initial runs showed that an acceptable sidelobe level is only achievable up to a certain aperture size.

To run this scenario a maximum aperture size and a maximum acceptable sidelobe level has to be entered by the user as a parameter.

If there are M transmitter and N receiver elements to be optimized this configuration yields M-1+N variables to play around with. Those variables have a range from zero to aperture size.

5.3.3 **Algorithms**

Random Descent

Random Descent algorithm –also mentioned as Reduced Variable Neighbourhood Search(RVNS)- is one of the variable neighbourhood search algorithms(VNS) [74]. Variable neighbourhood search algorithms start with one or more initial solutions. They then create a set of neighbours based on the initial solution(s) at hand. And depending on the algorithm a neighbour or more are selected and their fitness is evaluated. If the selected neighbour(s) yield better results than the initial solution, best one(s) of those evaluated solutions are chosen and whole process is iterated until either a time limit is fulfilled, an iteration limit is reached, or a good enough solution is found.

There are a few VNS methods usually diverging at how to create neighbours and which one to choose to evaluate [75]. Random descent or RVNS is possibly the simplest one to implement without getting stuck at local optima during search. Two example neighbourhood search algorithms that are likely to get stuck at local optimum are, first improvement neighbourhood search (first descent) and best improvement neighbourhood search (highest descent). The former algorithm simply chooses the first neighbour that yields an improvement, therefore not only it's likely but it's a fact that it's bound to get stuck. The latter algorithm evaluates a number of neighbourhoods and chooses the one that yields the highest improvement. It's only slightly different from the former but it's also bound to get stuck.

Random descent chooses its next neighbour completely randomly without a direction. This will cause the algorithm to sometimes miss the local optima but it allows it to jump over the neighbourhoods. On top of that, random descent algorithm can be improved by implementing a hybrid neighbourhood search algorithm, such as a combination of random descent and first descent. However this implementation is not in our scope since we are going to use smarter algorithms in either case.

The step-by-step case specific algorithm can be found below.

1. Initialize an empty solution set "good" solutions
2. Have an initial solution and set it as current solution
3. Evaluate fitness of current solution
    a. Generate pattern at zero degrees scan angle
    b. Compute beamwidth
    c. Compute sidelobe level
4. Save current solution as "best" solution
5. Randomly select one of the "good" solutions or the current solution to get a neighbour
6. Randomly select a neighbour by randomizing one of the randomly selected variables
7. Generate the neighbour's pattern and evaluate its fitness
    a. Generate pattern at zero degrees scan angle
    b. Compute beamwidth
    c. Compute sidelobe level

   d. Feasibility checks
      i. Check if grating lobes exist
      ii. Check if beamwidth is smaller than acceptable maximum beamwidth
      iii. Check if beam points to desired scan angle
      iv. Check if maximum relative sidelobe level is lower than maximum acceptable
      v. Check if edge sidelobe levels are lower than maximum acceptable
8. If fitness of neighbour is feasible AND better than current solution
   a. Add the current solution to the list of “good” solutions
   b. Save the neighbour as “best” solution
   c. Set the neighbour to be the current solution
9. If neighbour is NOT feasible OR NOT better than current solution
   a. Do nothing
10. Check if finishing conditions are met
   a. If finishing conditions are NOT met, go to step 5 and repeat
   b. If finishing conditions are met, stop
11. Return the current solution.

Simulated Annealing

As the name of the algorithm suggests, the idea behind this heuristic algorithm is to introduce entropy to the system and let it gradually cool down to remove stress[76]. The idea is taken from statistical thermodynamics; large systems with high entropy spontaneously approach to the equilibrium state (states with lower, perhaps lowest energy) [77].

This algorithm’s generic implementation is actually very similar to random descent; but unlike random descent there is still a chance that a neighbour might be accepted to be the current solution even if its fitness is not better. Ideally, in a simulated annealing algorithm there would be two solutions kept at each iteration; best solution and the current solution. In random descent these two are the same. In SA, current solution is just a state that is used to traverse solutions. This separation allows SA to get out of “bad” neighbourhoods where a local optimum is located and no neighbour of that local optimum offers a better fitness (therefore traps the current solution). In more detail, with a given probability, a neighbour will be accepted as current solution even if its fitness is lower than the current and/or best solution. And in the next iteration that current solution will probably be changed for a better solution than both solutions. So the “worse” solution will only have played the role of a bridge between the local optima and the better solutions.

However this is not the only difference from random descent. More importantly, a “cool-down” is implemented in SA, so that the algorithm doesn’t just bounce from one local optimum to another until execution is finished. Generally, the probability of acceptance depends on two things; temperature, and fitness difference. Temperature is a variable that defines the system’s entropy; it’s usually dependent on the current number of iteration or time. The fitness difference is a variable that denotes the difference between the current-solution and the neighbour. In summary, the acceptance probability is higher if the temperature is high (e.g. number of iteration or execution time is low). And acceptance probability is much lower if the neighbour is much worse than the current solution, but less low if the neighbour is slightly worse than the current solution.

Like many others, this algorithm and its variants have also been used to optimize phased array configurations and/or weighting coefficients of an array to obtain desirable patterns. For example [78] only focuses on weighting factors to obtain a desirable pattern for a given linear array configuration. However [21] includes element positions and their weights as variables for optimizing the array aperture size and sidelobe levels (similar to our optimization goals).

The step-by-step case specific algorithm can be found below.

1. Initialize two solutions; best solution and current solution
2. Have an initial solution and set it as current solution
3. Evaluate fitness of current solution
    a. Generate pattern at zero degrees scan angle
    b. Compute beamwidth
    c. Compute sidelobe level

4. Save current solution as best solution
5. Randomly select a neighbour by randomizing one of the randomly selected variables
6. Generate the neighbour's pattern and evaluate its fitness
    a. Generate pattern at scan angles of interest (i.e. sparsely separated scan angles to avoid heavy computing requirements, e.g. 0,15,30,45 degrees)
    b. Run feasibility checks for each scan angle
        i. Check if grating lobes exist
        ii. Check if beamwidth is smaller than acceptable maximum beamwidth
        iii. Check if beam points to desired scan angle
        iv. Check if maximum relative sidelobe level is lower than maximum acceptable
        v. Check if edge sidelobe levels are lower than maximum acceptable
    c. Generate pattern at zero degree scan angle
        i. Compute beamwidth
        ii. Compute sidelobe level
    d. Calculate fitness differences using the feasibility checks and optimization goal
        i. Neighbour's optimization goal subtracted from current solution's goal and kept as "delta"
        ii. Number of failed feasibility checks is multiplied with a constant and subtracted from "delta" (penalty for infeasible solutions).
7. If fitness difference is less than zero (i.e. a better solution is found)
    a. Accept the solution as current solution
    b. Calculate fitness difference with respect to best solution
    c. If current solution is better than best solution
        i. Save the current solution as best solution
8. If fitness difference is greater than zero (i.e. a worse solution is found)
    a. Calculate an acceptance probability parameter based on
        i. Current number of iteration
        ii. Fitness difference

   b. If generated random number is less than acceptance probability

      i. Accept the solution as current solution

9. Increment the number of iteration, decrease the temperature of the system
10. If temperature is not zero, go to step 5.
11. Return the best solution

Genetic Algorithm

Genetic algorithm, as might be guessed from the name incorporates pseudo-biology as layer to the method between problem and its solution. In a very quick summary; genetic algorithm is a heuristic algorithm where solutions are represented as genes (e.g. bit vectors), and genetic operations of nature (mutation, crossover, immigration, survival of the fittest, etc.) are used to evolve these genes into better solutions. It usually involves more than a single solution at each iteration -i.e. a population- eventually evolving to generate the fittest solution (individual). Idea is taken from the nature itself, where aforementioned operations already exist and the –current- result is the human beings. Somewhat relevantly, the idea that Earth itself is a biological computer running a very advanced genetic algorithm to solve a very difficult problem has even found itself a place in the popular culture [79].

The history of genetic algorithms go as back as 1950 with Alan Turing's proposal of a learning machine which would work with the principles of evolution [80]. The term "genetic algorithm" however was first coined by Holland [81]. The subject of genetic algorithm has since evolved itself to subjects such as evolution strategies, evolutionary programming, artificial life, classifier systems, genetic programming and evolvable hardware [82]. These sub-categories will not be discussed here. The very details and detailed history of genetic algorithm also will not be discussed any further. But for detailed explanations one can refer to [81] as the originating source, or [82] for a more recent source.

A genetic algorithm is very different than other heuristic algorithms mentioned. And some keywords or definitions might be different than previously mentioned. In a genetic algorithm, solutions are also often referred as *individuals* or *chromosomes*. And solution sets can be referred as *populations* due to analogy to biology. Iterations are also often called *generations*. This algorithm is rather an analogy to the biological evolution process, therefore operations on solutions are also generally different than previously mentioned. Generically all genetic algorithms share a specific structure of iterations [82].

1. Initialize a population
2. Select parent chromosomes
3. Perform crossover (mating of parents)
4. Perform mutation (depending on a probability)
5. Evaluate fitness of new individuals (and save best)
6. Stop, if matching criteria is reached or predefined number of generations is reached
7. Else go to step 2

The methods of performing these steps vary a lot. And sometimes there are extra steps to take such as inversion, immigration, hyper-mutation, preservation of best individual, etc. Also depending on the implementation, the population size might be growing, maintained, or simply be variable to allow diversity.

As one can notice, unlike other previously mentioned algorithms, these steps don't include any control mechanism to take the population to a global optimum let alone a local optimum. Therefore in cases with small populations or relatively low number of generations it is entirely possible not to obtain a single good solution or even a feasible one. So, when using a genetic algorithm implementation, the author can generally hope to have a large enough population and have enough generations to eventually reach a good solution. But, with the increased computational power, this argument is becoming less of an issue as shown by the results of this research too.

The uses of genetic algorithm in sparse linear array optimization has already been mentioned in [69][20][70] and [71]. As a not previously mentioned but rather short and easy-to-grasp source for sparse phased array pattern synthesis can be found in [83]. Genetic algorithm has also already been used for MIMO optimization in different cases too; a sidelobe level optimization case for through the wall imaging purposes [84]. And there is another one similar to our case of finding an optimized pattern for a thinned MIMO array [45]. However, that research considers relatively high number of elements for its sub-arrays such as 16-by-16 MIMO arrays, and is not an experimental research.

The step-by-step case specific algorithm can be found below.

1. Initialize a fixed size population
    a. Half randomized
    b. Half obtained from other algorithms
2. Run a generation:
    a. Decode the population from bit sequences to actual variables
    b. Evaluate the fitness of individuals
        i. Generate pattern at multiple degrees of scan angles (i.e. sparsely separated to avoid heavy computing requirements, e.g. 0,15,30,45 degrees)
        ii. Run feasibility checks for each scan angle
            1. Check if grating lobes exist
            2. Check if beamwidth is smaller than acceptable maximum beamwidth
            3. Check if beam points to desired scan angle
            4. Check if maximum relative sidelobe level is lower than maximum acceptable
            5. Check if edge sidelobe levels are lower than maximum acceptable
        iii. Compute a score from optimization goal and the feasibility checks
    c. Save the individual with the highest fitness as the best individual.
    d. Until same population size is reached;
        i. Select two parents(individuals) with tournament selection (higher the fitness, higher the probability of being chosen)
        ii. With a given probability
            1. Swap parts of the bit sequences from a random index and copy the new individuals into the new population (Crossover)
            2. OR directly copy the parents into the new population

iii. With a given probability inverse a random bit from the new individuals (Mutation)

e. Copy the best individual from the previous population into the new population to preserve the best fitness

3. If predefined number of generations is NOT reached; go to step 2.
4. Evaluate the fitness of the last population
5. If there is a solution with better fitness than the previously saved best individual
    a. Return the new best individual
6. Else;
    a. Return the previously saved best individual

5.3.4 Sample Optimization Process

To further explain what optimization algorithms' outputs look like and what they perform over time, states from beginning, halfway and end of optimization algorithm is presented below in Figure 5-1, Figure 5-2 and Figure 5-3 respectively. The presented algorithm is Simulated Annealing and the goal is beamwidth minimisation. On the right chart, it can even be observed how system temperature drops over time and the mean of the beamwidth is in a downward trend.

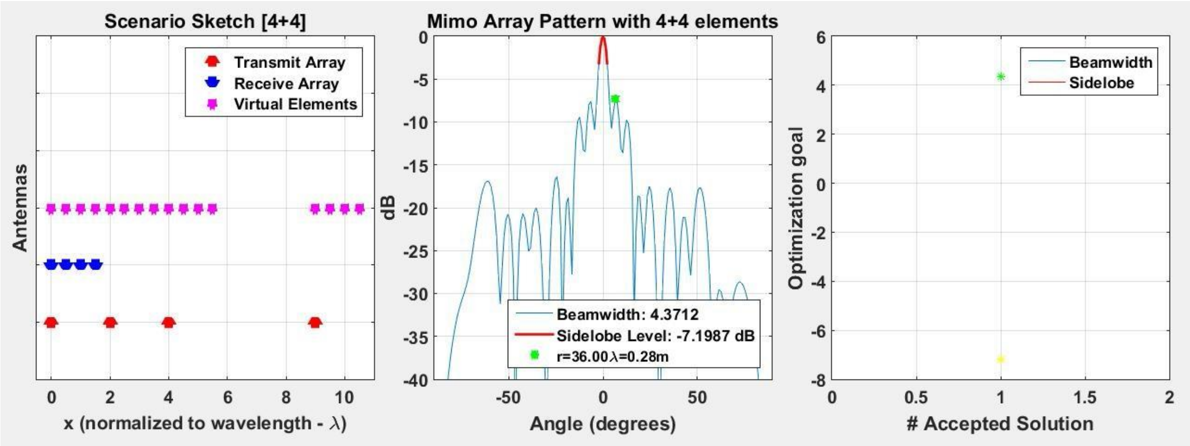


Figure 5-1 Beginning of optimization. No solutions yet.

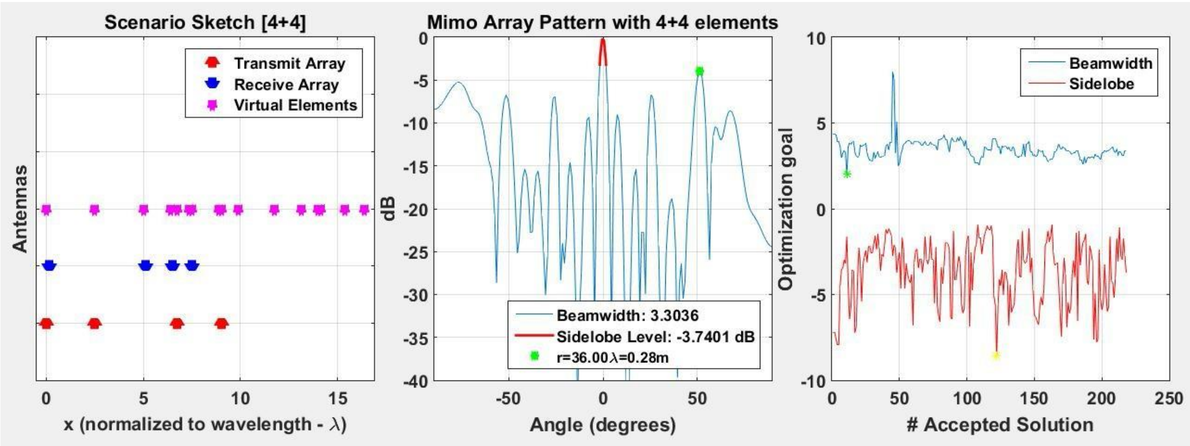


Figure 5-2 Halfway through the optimization. Some solutions exist, but nothing extraordinary

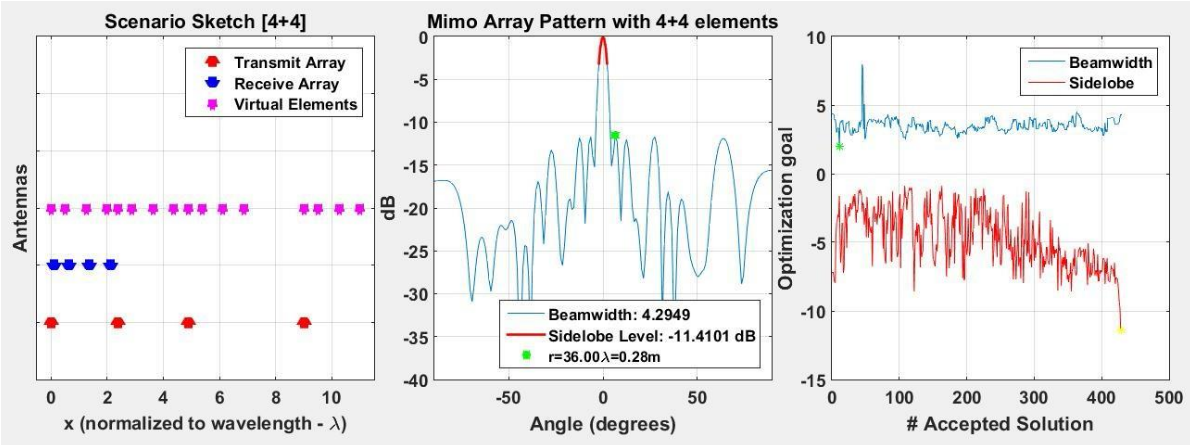


Figure 5-3 End of the optimization. Final solution with a relatively very-low sidelobe level is found and accepted.

5.4 OPTIMISATION RESULTS

Implemented algorithms were run with similar parameters for both types of optimizations goals and some of the usable array configurations obtained via these algorithms are presented. Note that these algorithms have been run more than once and the best results from these runs are hand-picked and displayed.

5.4.1 Random Descent

Below in Table 5-1 is the explanations of the parameters as they're defined in code. After that is Table 5-2 which contains the shared values for the parameters used to run the random descent algorithm to generate array configurations.

Table 5-1 Random descent optimisation parameters and their definitions

Parameter Name	Definition
Optimization_type	Defines the optimization goal; minimum beamwidth or minimum sidelobe
Step_size	Defines the minimum step size when moving antenna elements.
Maximum_aperture_size	Defines the fixed aperture size
Finishing_condition	Defines the stopping conditions
Num_iterations	Defines the number of iterations to stop the optimization
Iterations_time	Defines the time needed to optimize
Scan_angles_degree	A vector of scan angles in degrees to generate patterns and check. The size of this vector directly affects the speed of iterations.
Scan_capability_maximum_scan_angle	Defines the maximum scan angle to check for scan capability

Scan_capability_step_size	Check scanning capability in steps of this number starting from zero
Optimization_range	Defines the range that the array is trying to operate at with respect to targets
Max_beamwidth	Defines the maximum tolerable beamwidth for minimum sidelobe optimizations
Max_tolerable_sidelobe_level	Defines the maximum tolerable sidelobe level for minimum beamwidth optimizations
Edge_sidelobe_level	Defines the preferable sidelobe level at the pattern edges
Beam_direction_error_margin	Defines the tolerable error in beam direction
Penalty_invalid_edge_level	Defines the penalty for not satisfying the edge_sidelobe_level value
Penalty_invalid_sidelobe_level	Defines the penalty for not satisfying the maximum_tolerable_sidelobe_level
Penalty_invalid_beam_direction	Defines the penalty for not satisfying the Beam_direction_error_margin
Penalty_invalid_beamwidth	Defines the penalty for not satisfying the max_tolerable_beamwidth
Penalty_grating_lobe	Defines the penalty for having grating lobes

When optimisation type is chosen to be fixed aperture size, the optimisation goal becomes minimising the sidelobe levels. Whereas if the type is chosen to be variable aperture size, the optimisation goal becomes minimising the beamwidth. Therefore cost function (optimisation goal) for random descent is defined rather simple as the difference between the best found beamwidth and the current beamwidth, added with the penalties from feasibility checks.

$$delta = beamwidth_{current} - beamwidth_{best} + penalties \qquad 5\text{-}1$$

The cost function for minimising beamwidth can be summarised as in 5-1, in which delta is tried to be minimised. And the acceptance probability in random descent simply depends on this *delta* being negative or positive. So assuming there are no penalties, if the current beamwidth is lower than best beamwidth, it will cause delta to be less than zero and the current solution will be accepted. If the current beamwidth is bigger than the best beamwidth, then delta will become positive and the current solution will not be accepted.

It follows a similar logic when the optimisation type fixed aperture size. In this case, cost function (optimisation goal) for random descent is defined as the difference between the best found sidelobe level and the current sidelobe level, added with the penalties from feasibility checks.

$$delta = sidelobe_{current} - sidelobe_{best} + penalties \qquad 5\text{-}2$$

The cost function for minimising sidelobe level can be summarised as in 5-2, in which delta is tried to be minimised. And the acceptance probability in random descent simply depends on this *delta* being negative or positive. So assuming there are no penalties, if the current sidelobe level is lower than best sidelobe level, it will cause delta to be less than zero and the current solution will be accepted. If the current sidelobe level is bigger than the best sidelobe level, then delta will become positive and the current solution will not be accepted.

An important thing to note is that if the beamwidth or sidelobe differences between solutions would be relatively small (i.e. less than 1) in a neighbourhood, then any penalty higher than 1 is going to immediately cause a solution to be rejected. The differences between solutions in a neighbourhood and the quality of the solution in a neighbourhood is highly dependent on the optimisation parameters. Below presented parameters in Table 5-2 are chosen after a number of trials to yield the seemingly best results. Note that any text in italic in value column is a reference to a variable or a named constant also defined in the same table.

Table 5-2 Random descent parameters

Parameter Name	Possible Values/unit	Value

Optimization_type	*Variable_aperture_size*; *Fixed_aperture_size*	Not shared between different optimization types
Step_size	Positive real numbers in terms of wavelength	0.05
Maximum_aperture_size	Positive real numbers in terms of wavelength	9
Finishing_condition	*Finish_on_time; finish_on_iterations; finish_on_first; finish_on_both*	*Finish_on_time*
Num_iterations	Positive integer	3e3
Iterations_time	Positive real number in minutes	2
Scan_angles_degree	Vector of real numbers in degrees	[-89.9:0.1:89.9]
Scan_capability_maximum_scan_angle	Positive real number in degrees	30
Scan_capability_step_size	Positive real number in degrees	15
Optimization_range	Positive real number in terms of wavelength	3**Maximum_aperture_size*
Max_beamwidth	Positive real number in degrees	10
Max_tolerable_sidelobe_level	Negative real number in decibels	*Not shared between optimization types*

Edge_sidelobe_level	Negative real number in decibels	*Max_tolerable_sidelobe_level* – 1
Beam_direction_error_margin	Real positive number in degrees	0.5
Penalty_invalid_edge_level	Real number, multiplier; unitless	0
Penalty_invalid_sidelobe_level	Real number, multiplier; unitless	3
Penalty_invalid_beam_direction	Real number, multiplier; unitless	5
Penalty_invalid_beamwidth	Real number, multiplier; unitless	1
Penalty_grating_lobe	Real number, multiplier; unitless	3

The finishing condition was chosen to be *finish_on_time*, because a time limitation was found to be more logical to finish trials rather than a set of iterations, because it would put a hard limit on computation time regardless of the platform's computation power. The parameter *Scan_capability_maximum_scan_angle* was chosen to be 30 degrees, since this corresponds to the maximum beamwidth of the sensors available. The parameter *optimization_range* was chosen to be a relative triple of *maximum_aperture_size* which is a relatively difficult near-field range to be able to form beams. The penalties at the bottom were selected via number of trials; these values were selected regarding the importance of the penalties with respect to each other and relative damage that we chose to impose on the fitness of a solution.

Fixed Aperture Size

In fixed aperture size optimization, two edge transmitter locations are fixed to ensure fixed physical aperture size. However the remaining elements (both the transmitters and the receivers) have the freedom to move around any point in between those elements. Below is the list of names and respective values for non-shared parameters for this type of optimization.

- **Optimization_type:** Fixed_aperture_size
- **Max_tolerable_sidelobe_level:** -6 dB

A maximum tolerable sidelobe level of -6 dB was chosen so that the initial jumps wouldn't cause the algorithm to get stuck in a bad neighbourhood. Since this algorithm runs with random processes the results below are simply an example to what could be achieved. A summary of the results, plots of patterns at two different scan angles and a sketch of the array configuration can be found below. In Table 5-3 one of the array configurations obtained via running the random descent optimisation algorithms can be found, along with its computed sidelobe level and computed beamwidth.

Table 5-3 Sample optimised MIMO array with the use of random descent with fixed aperture size

	Value(s)	**Unit**
Transmit Antenna Positions	[0, 2.1, 4, 9]	Wavelength
Receive Antenna Positions	[0.35, 1, 1.5, 4.2]	Wavelength
Sidelobe Level	-10.37	dB
Beamwidth	4.12	Degrees

In Figure 5-4 below, the array configuration's top view sketch can be found in detail. On top left the transmitter array positioning and on top right receiver array positioning can be seen separately. In addition, in the middle the superposed transmitter and receiver configuration the algorithm found originally can be observed. Finally, on bottom the virtual array element distribution can be observed.

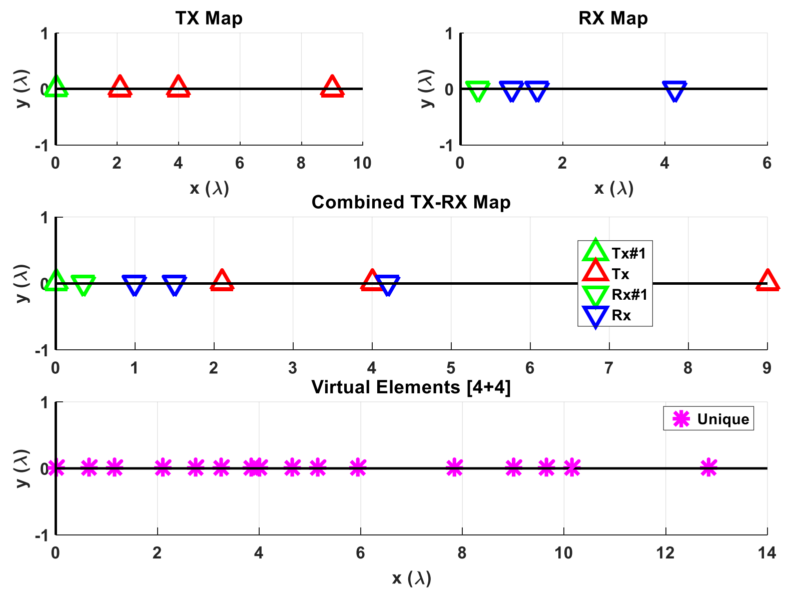


Figure 5-4 Exemplary random descent fixed size array optimization configuration result

In Figure 5-5 the computed beam pattern can be seen, if such array was scanning at 0 degrees. The measured beamwidth is 4.12 degrees which is about 1.5 degrees less than a 5.56 degrees beamwidth expected from an array same size. And the sidelobe level is measured as -10.72 dB which is about 3 dB higher than the standard -13.1 sidelobe level.

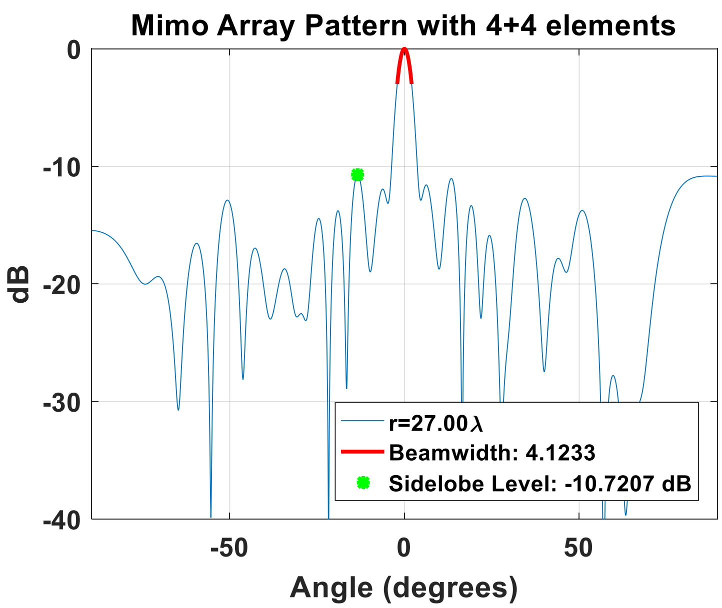


Figure 5-5 Exemplary random descent algorithm result scanning at zero degrees

In Figure 5-6 the computed beam pattern can be seen, if such array was scanning at 30 degrees. The measured beamwidth is 4.69 degrees which is about 1.5 degrees less than a 6.42 degrees beamwidth expected from an array same size. And the sidelobe level is measured as -6.06 dB which is about 7 dB higher than the standard -13.1 sidelobe level.

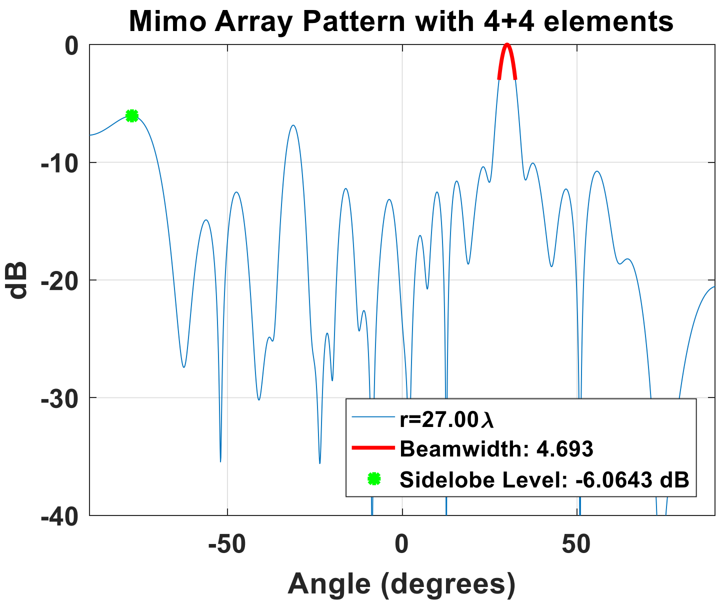


Figure 5-6 Exemplary random descent algorithm result scanning at 30 degrees

Variable Aperture Size

In variable aperture size optimization only one edge transmitter location is fixed as a reference point and to ensure that during random variations similar configurations are skipped. The remaining elements (both the receivers and the transmitters) have the freedom to position up to a predefined maximum aperture size. Below is the list of names and respective values for non-shared parameters for this type of optimization.

- **Optimization_type:** Variable_aperture_size
- **Max_tolerable_sidelobe_level:** -10 dB

Contrary to the other optimisation type a higher maximum tolerable sidelobe level has been chosen, because this optimisation goal has proven itself to really push the aperture size to obtain narrow beamwidths which eventually causes the sidelobe levels to go down. Since this algorithm runs with random processes the results below are simply an example to what could be achieved. A summary of the results, plots of patterns at two different scan angles and a sketch of the array configuration can be found below. In Table 5-4, one of the array configurations obtained via running the random descent optimisation algorithms can be found, along with its computed sidelobe level and computed beamwidth.

Table 5-4 Sample optimised MIMO array with the use of random descent with variable aperture size

	Value(s)	**Unit**
Transmit Antenna Positions	[0, 2, 4, 6]	Wavelength
Receive Antenna Positions	[0, 2.5, 3.05, 5.55]	Wavelength
Sidelobe Level	-10.14	dB
Beamwidth	5.02	Degrees

In Figure 5-7 below, the array configuration's top view sketch can be found in detail. On top left the transmitter array positioning and on top right receiver array positioning can be seen separately. In addition, in the middle the superposed transmitter and receiver configuration the algorithm found originally can be observed. Finally, on bottom the virtual array element distribution can be observed.

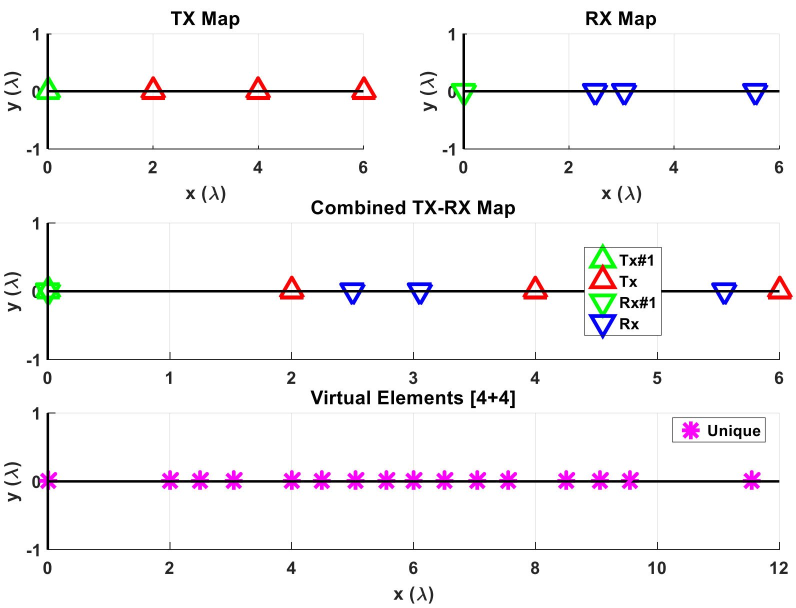


Figure 5-7 Exemplary random descent variable size array optimization configuration result

In Figure 5-8 the computed beam pattern can be seen, if such array was scanning at 0 degrees. The measured beamwidth is 5.02 degrees which is about 3 degrees less than a 8.33 degrees beamwidth expected from an array same size. And the sidelobe level is measured as -10.14 dB which is about 3 dB higher than the standard -13.1 sidelobe level.

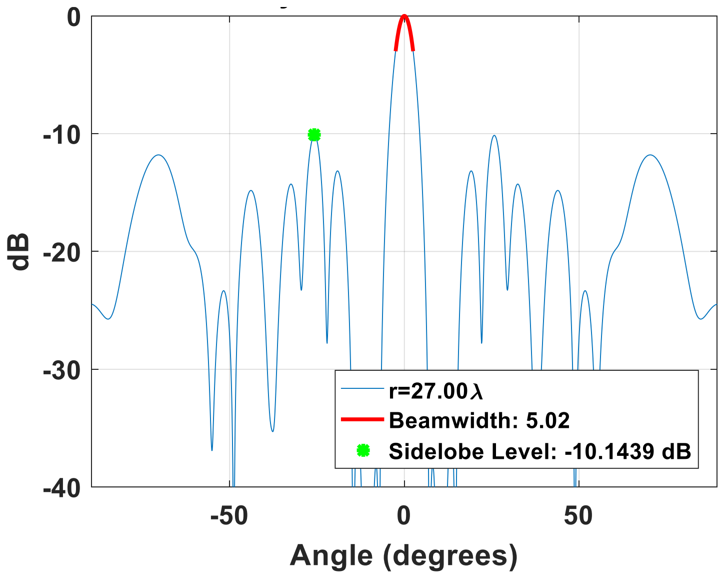


Figure 5-8 Exemplary random descent algorithm result scanning at zero degrees

In Figure 5-9 the computed beam pattern can be seen, if such array was scanning at 30 degrees. The measured beamwidth is 5.77 degrees which is about 4 degrees less than a 9.62 degrees beamwidth expected from an array same size. And the sidelobe level is measured as -10.17 dB which is about 3 dB higher than the standard -13.1 sidelobe level.

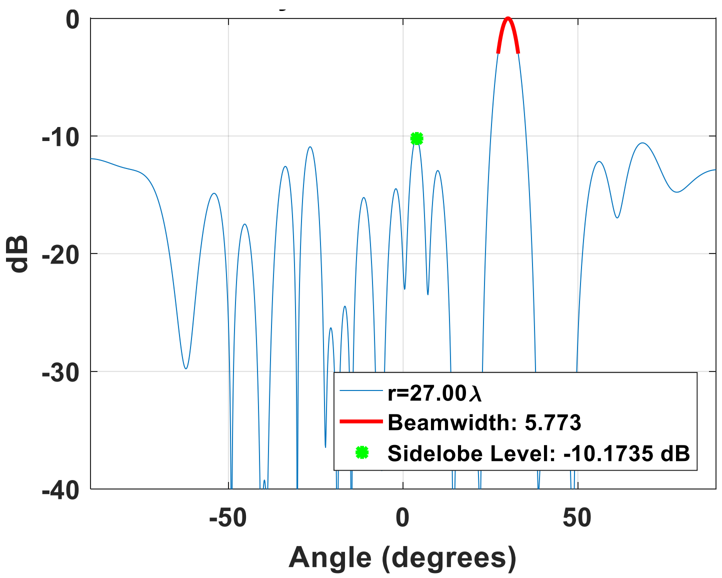


Figure 5-9 Exemplary random descent algorithm result scanning at 30 degrees

Both these results already demonstrate that even with random descent, improvements can be obtained. However, it was more likely to find even better configurations with more advanced search algorithms. So these configurations were not experimentally tested, but only saved for future use.

5.4.2 Simulated Annealing

Below in Table 5-5 is the explanations of the parameters as they're defined in code. After that is Table 5-6 which contains the shared values for the parameters used to run the simulated annealing algorithm to generate array configurations.

Table 5-5 Simulated annealing parameters and their definitions

Parameter Name	Definition
Optimization_type	Defines the optimization goal; minimum beamwidth or minimum sidelobe
Step_size	Defines the minimum step size when moving antenna elements.
Acceptance_prob_multiplier	Used to convert temperature to meaningful probabilities i.e. $e^{\left(-\frac{delta*ACCEPTANCE\_PROB\_MULTIPLIER}{currentTemperature}\right)}$
Initial_temperature	System's virtual initial temperature. System should annealing super-hot to allow interesting solutions
Final_temperature	Temperature to stop annealing, i.e. room temperature.
Temperature_drop_multiplier	Defines how fast/slow the temperature drops. Theoretically slower temperature drops should yield better results, which also means more iterations
Num_iterations_per_temperature	Number of iterations to be done at a given temperature level
Maximum_aperture_size	Defines the fixed aperture size
Scan_angles_degree	A vector of scan angles in degrees to generate patterns and check. The size of this vector directly affects the speed of iterations.
Scan_capability_maximum_scan_angle	Defines the maximum scan angle to check for scan capability

Scan_capability_step_size	Check scanning capability in steps of this number starting from zero
Optimization_range	Defines the range that the array is trying to operate at with respect to targets
Max_beamwidth	Defines the maximum tolerable beamwidth for minimum sidelobe optimizations
Max_tolerable_sidelobe_level	Defines the maximum tolerable sidelobe level for minimum beamwidth optimizations
Edge_sidelobe_level	Defines the preferable sidelobe level at the pattern edges
Beam_direction_error_margin	Defines the tolerable error in beam direction
Penalty_invalid_edge_level	Defines the penalty for not satisfying the edge_sidelobe_level value
Penalty_invalid_sidelobe_level	Defines the penalty for not satisfying the maximum_tolerable_sidelobe_level
Penalty_invalid_beam_direction	Defines the penalty for not satisfying the Beam_direction_error_margin
Penalty_invalid_beamwidth	Defines the pnelty for not satisfying the max_tolerable_beamwidth
Penalty_grating_lobe	Defines the penalty for having grating lobes

When optimisation type is chosen to be fixed aperture size, the optimisation goal becomes minimising the sidelobe levels. Whereas if the type is chosen to be variable aperture size, the optimisation goal becomes minimising the beamwidth. Therefore cost function (optimisation goal) for simulated annealing is also defined as the difference between the previous solutions's beamwidth and the current beamwidth, added with the penalties from feasibility checks. Note that, unlike random descent comparison is only made with respect to previous accepted solution rather than the best solution found.

$$delta = beamwidth_{current} - beamwidth_{previous} + penalties \quad 5\text{-}3$$

$$probability_{acceptance} = e^{\left(-\frac{delta * ACCEPTANCE\_PROB\_MULTIPLIER}{temperature_{current}}\right)} \quad 5\text{-}4$$

$$temperature_{current} = TEMPERATURE\_DROP\_MULTIPLIER * temperature_{current} \quad 5\text{-}5$$

The cost function for minimising beamwidth can be summarised as in 5-3, in which delta is tried to be minimised. But unlike random descent the acceptance probability in simulated annealing depends on the system's "temperature", which is a function of *delta* as given in 5-4. The system temperature on the other hand is a pre-defined value that exponentially decreases with respect to a pre-defined number of iterations. The way it is decreased at every iteration is given in 5-5. So assuming there are no penalties, if the current beamwidth is lower than best beamwidth, it will cause delta to be less than zero and the current solution will be immediately accepted. However -unlike random descent- if the delta is greater than zero it will then generate an acceptance probability less than 1. Then a random number will be compared to the acceptance probability and if it's less than the acceptance, the solution will still be accepted. If the random number is higher than acceptance then it's rejected. So there's still some chance that an accepted solution may not be better than the current solution.

It follows a similar logic when the optimisation type fixed aperture size. In this case, cost function (optimisation goal) for simulated annealing is defined as the difference between previous solution's sidelobe level and the current sidelobe level, added with the penalties from feasibility checks.

$$delta = sidelobe_{current} - sidelobe_{previous} + penalties \quad 5\text{-}6$$

The cost function for minimising sidelobe can be summarised as in 5-6, in which delta is tried to be minimised. So assuming there are no penalties, if the current sidelobe level is lower than best sidelobe level, it will cause delta to be less than zero and the current solution will be immediately accepted. However if the delta is greater than zero it will then generate an acceptance probability less than 1. Then a random number will be compared to the acceptance probability and if it's less than the acceptance, the solution will still be accepted. If the random number is higher than acceptance then it's rejected. So there's still some chance that an accepted solution may not be better than the current solution.

An important thing to note is that if the beamwidth or sidelobe differences between solutions would be relatively small (i.e. less than 1) in a neighbourhood, then any penalty high enough is going to immediately cause the acceptance probability to drop to very low levels; effectively rejecting them. Below presented parameters in Table 5-6 are chosen after a number of trials to yield the seemingly best results. Note that any text in italic in value column is a reference to a variable or a named constant also defined in the same table.

Table 5-6 Simulated annealing parameters

Parameter Name	Possible Values/unit	Value
Optimization_type	Variable_aperture_size ; Fixed_aperture_size	<u>Not shared between different optimization types</u>
Step_size	Positive real numbers in terms of wavelength	0.125
Acceptance_prob_multiplier	Positive real number, unitless	1e4
Initial_temperature	Positive real number, unitless	1320503
Final_temperature	Non-zero, positive real number, unitless	200
Temperature_drop_multiplier	Positive real number, less than 1	0.995
Num_iterations_per_temperature	Positive real integer	1
Maximum_aperture_size	Positive real numbers in terms of wavelength	9
Scan_angles_degree	Vector of real numbers in degrees	[-89.9:0.1:89.9]

Scan_capability_maximum_scan_angle	Positive real number in degrees	30
Scan_capability_step_size	Positive real number in degrees	15
Optimization_range	Positive real number in terms of wavelength	3**Maximum_aperture_size*
Max_beamwidth	Positive real number in degrees	10
Max_tolerable_sidelobe_level	Negative real number in decibels	<u>*Not shared between optimization types*</u>
Edge_sidelobe_level	Negative real number in decibels	*Max_tolerable_sidelobe_level* – 1
Beam_direction_error_margin	Real positive number in degrees	0.5
Penalty_invalid_edge_level	Real number, multiplier; unitless	0
Penalty_invalid_sidelobe_level	Real number, multiplier; unitless	3
Penalty_invalid_beam_direction	Real number, multiplier; unitless	5
Penalty_invalid_beamwidth	Real number, multiplier; unitless	1
Penalty_grating_lobe	Real number, multiplier; unitless	3

Since this algorithm really is an analogy to an annealing system, the parameter choice was mostly done on a trial and error basis. The parameter *Initial_temperature* was chosen 1320503 to allow a sufficiently high entropy for the system to allow all kinds of solutions to be accepted initially; effectively almost making the algorithm run worse than random descent. Yet *temperature_drop_multiplier* was chosen as high as 0.995 so that temperature wouldn't drop too quickly and turn into random descent, and still would be able to jump between bad neighbourhoods. The parameter *acceptance_prob_multiplier* was chosen as 1e4 to be able to get the computed *delta*s to a proper range of values to have acceptance probabilities not biased towards accepting or rejecting all the time.

Fixed Aperture Size

In fixed aperture size optimization, two edge transmitter locations are fixed to ensure fixed physical aperture size. However the remaining elements (both the transmitters and the receivers) have the freedom to move around any point in between those elements. Below is the list of names and respective values for non-shared parameters for this type of optimization.

- **Optimization_type:** Fixed_aperture_size
- **Max_tolerable_sidelobe_level:** -6 dB

A maximum tolerable sidelobe level of -6 dB was chosen so that the initial jumps wouldn't cause the algorithm to get stuck in a bad neighbourhood. Since this algorithm runs with random processes the results below are simply an example to what could be achieved. A summary of the results, plots of patterns at two different scan angles and a sketch of the array configuration can be found below. In Table 5-7 one of the array configurations obtained via running the random descent optimisation algorithms can be found, along with its computed sidelobe level and computed beamwidth.

Table 5-7 Sample optimised MIMO array with the use of random descent with variable aperture size

	Value(s)	**Unit**
Transmit Antenna Positions	[0, 2.125, 4, 9]	Wavelength
Receive Antenna Positions	[0.375, 3, 3.625, 4.25]	Wavelength
Sidelobe Level	-10.91	dB
Beamwidth	4.12	Degrees

In Figure 5-10 below, the array configuration's top view sketch can be found in detail. On top left the transmitter array positioning and on top right receiver array positioning can be seen separately. In addition, in the middle the superposed transmitter and receiver configuration the algorithm found originally can be observed. Finally, on bottom the virtual array element distribution can be observed.

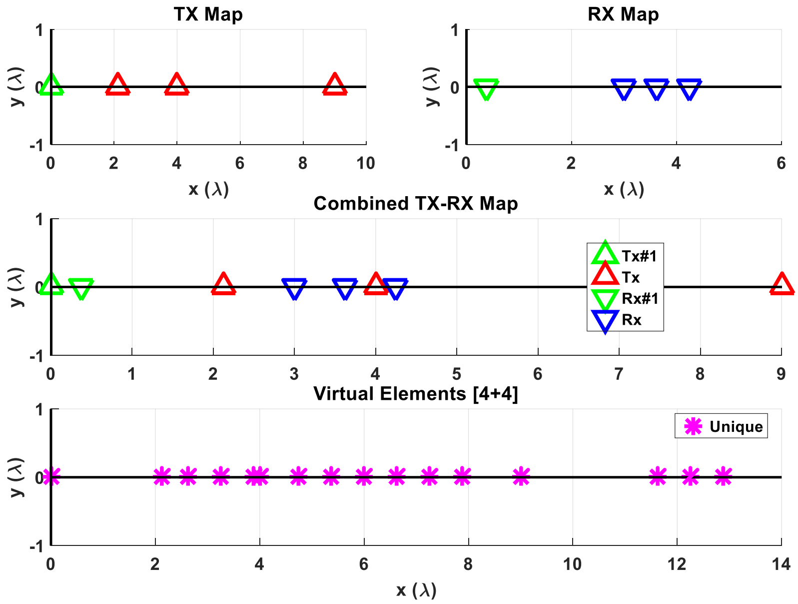


Figure 5-10 Exemplary simulated annealing fixed size array optimization configuration result

In Figure 5-11 the computed beam pattern can be seen, if such array was scanning at 0 degrees. The measured beamwidth is 4.12 degrees which is about 1.4 degrees less than a 5.56 degrees beamwidth expected from an array same size. And the sidelobe level is measured as -10.87 dB which is about 3 dB higher than the standard -13.1 sidelobe level.

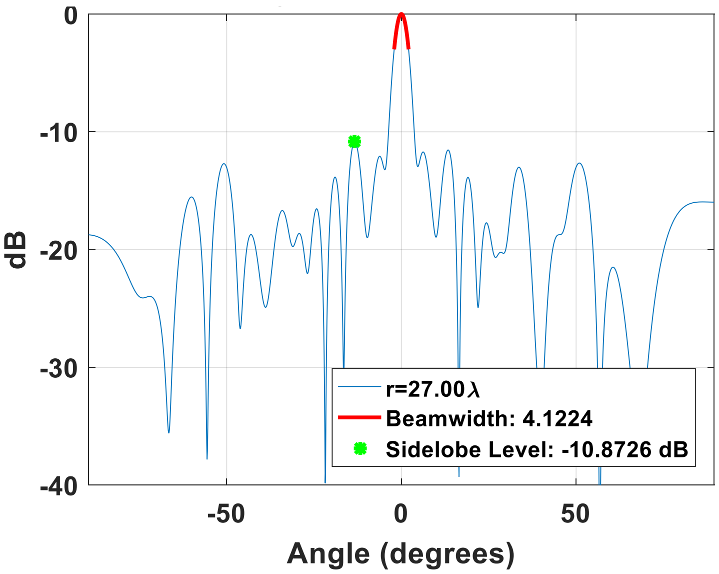


Figure 5-11 Exemplary simulated annealing algorithm result scanning at zero degrees

In Figure 5-12 the computed beam pattern can be seen, if such array was scanning at 30 degrees. The measured beamwidth is 4.70 degrees which is about 1.8 degrees less than a 6.42 degrees beamwidth expected from an array same size. And the sidelobe level is measured as -4.33 dB which is about 8 dB higher than the standard -13.1 sidelobe level.

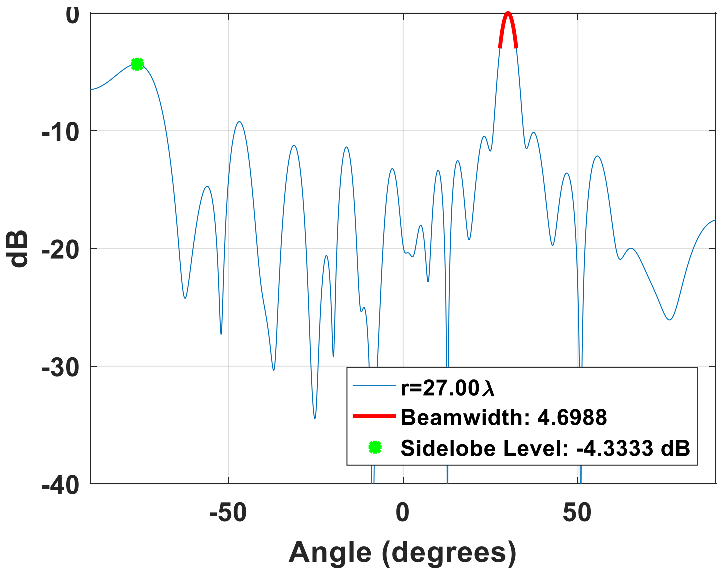


Figure 5-12 Exemplary simulated annealing algorithm result scanning at 30 degrees

Variable Aperture Size

In variable aperture size optimization, one edge transmitter location is fixed to ensure varying physical aperture size with minimum redundant configurations with same aperture size. However the remaining elements (both the transmitters and the receivers) have the freedom to move around any point in between those elements. Below is the list of names and respective values for non-shared parameters for this type of optimization.

- **Optimization_type:** Variable_aperture_size
- **Max_tolerable_sidelobe_level:** -10 dB

Contrary to the other optimisation type a higher maximum tolerable sidelobe level has been chosen, because this optimisation goal has proven itself to really push the aperture size to obtain narrow beamwidths which eventually causes the sidelobe levels to go down. Unfortunately, after many attempts simulated annealing approach did not yield very good solutions for variable aperture size optimization. The algorithm traverses through solutions that provide very good beamwidths but not so good sidelobe levels. Almost none that can hold a -10 dB sidelobe level while scanning. But, it’s not impossible.

Below is an example configuration found via simulated annealing with some beamwidth optimisation. A summary of this result, plots of patterns at two different scan angles and a sketch of the array configuration can be found below. In Table 5-8, one of the array configurations obtained via running the random descent optimisation algorithms can be found, along with its computed sidelobe level and computed beamwidth.

Table 5-8 Sample optimised MIMO array with the use of random descent with fixed aperture size

	Value(s)	**Unit**
Transmit Antenna Positions	[0, 2, 4, 7.125]	Wavelength
Receive Antenna Positions	[0, 0.5, 1, 1.5]	Wavelength
Sidelobe Level	-11.26	dB
Beamwidth	5.49	Degrees

In Figure 5-13 below, the array configuration's top view sketch can be found in detail. On top left the transmitter array positioning and on top right receiver array positioning can be seen separately. In addition, in the middle the superposed transmitter and receiver configuration the algorithm found originally can be observed. Finally, on bottom the virtual array element distribution can be observed.

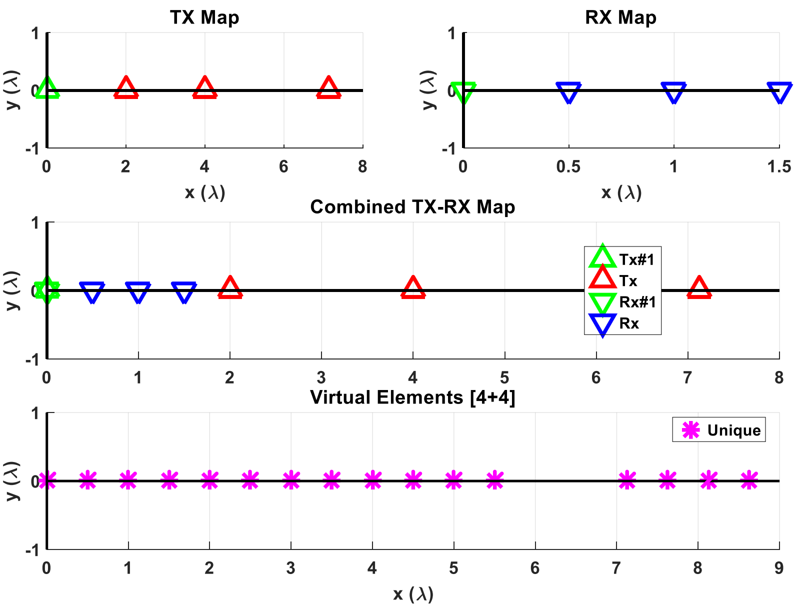


Figure 5-13 Exemplary simulated annealing variable size array optimization configuration result

In Figure 5-14 the computed beam pattern can be seen, if such array was scanning at 0 degrees. The measured beamwidth is 5.49 degrees which is about 1.5 degrees less than a 7.02 degrees beamwidth expected from an array same size. And the sidelobe level is measured as -11.26 dB which is about 2 dB higher than the standard -13.1 sidelobe level.

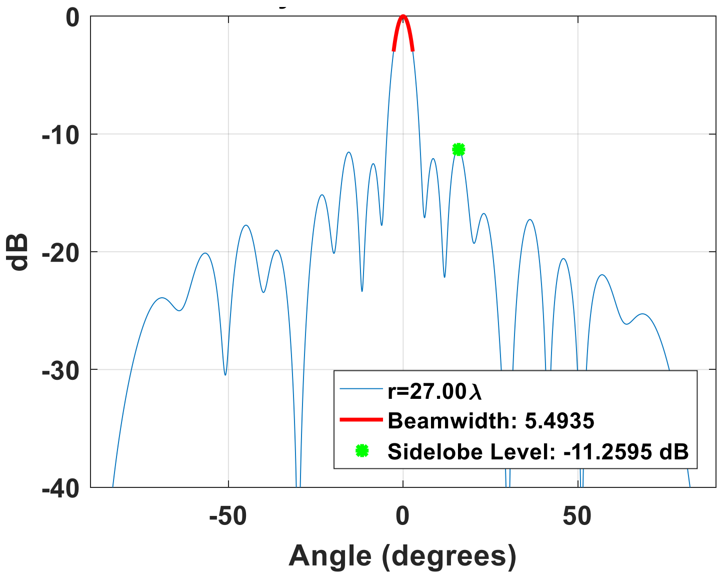


Figure 5-14 Exemplary simulated annealing algorithm result scanning at zero degrees

In Figure 5-15 the computed beam pattern can be seen, if such array was scanning at 30 degrees. The measured beamwidth is 6.30 degrees which is about 1.8 degrees less than a 8.10 degrees beamwidth expected from an array same size. And the sidelobe level is measured as -10.62 dB which is about 2.5 dB higher than the standard -13.1 sidelobe level.

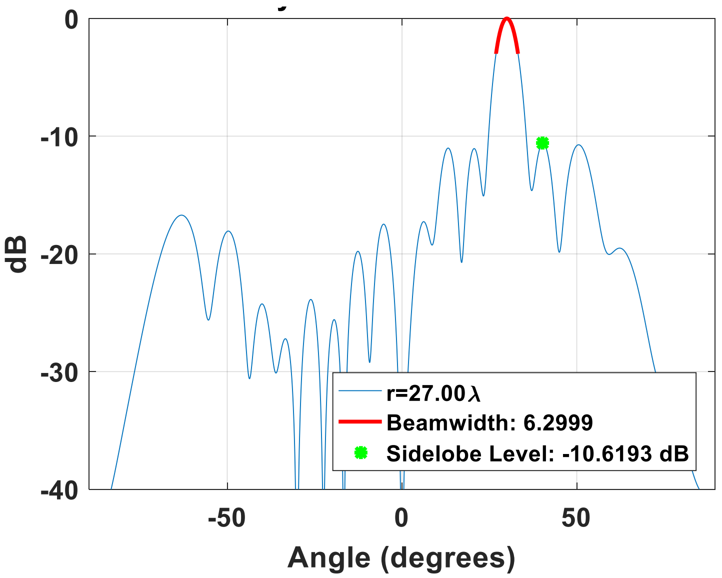


Figure 5-15 Exemplary simulated annealing algorithm result scanning at 30 degrees

Both these results demonstrate that improvements can be obtained with simulated annealing. However, it is more likely to find even better configurations with more advanced search algorithms. So these configurations were also not experimentally tested, but only saved for future use.

5.4.3 Genetic Algorithm

Below in Table 5-9 is the explanations of the parameters as they're defined in code. After that is Table 5-10 which contains the shared values for the parameters used to run the genetic algorithm to generate array configurations.

Table 5-9 Genetic algorithm parameters and their definitions

Parameter Name	Definition
Optimization_type	Defines the optimization goal; minimum beamwidth or minimum sidelobe
loadedPopulationSize	Defines the initial population size that is loaded from a data file, usually from results of other heuristics
randomizedPopulationSize	Defines the initial population size that is completely randomized
numberOfVariables	Number of variables to be optimized, i.e. antenna positions
numberOfGenes	This is an absolute value that defines the total number of genes, this variable adjusts the bits per variable
mutationProbability	Defines the probability of a mutation after a crossover
tournamentSelectionParameter	This parameter is used to calculate probabilities of selection during a tournament, this is the probability of choosing the best individual from the tournament pool
numberOfGenerations	Define the generations to run the genetic algorithm for. This is directly proportional to the execution time
tournamentSize	Defines how many individuals will form a tournament pool
numberOfReplications	Defines how many replicas of the best individual will be copied to new generation
Maximum_aperture_size	Defines the fixed aperture size

Scan_angles_degree	A vector of scan angles in degrees to generate patterns and check. The size of this vector directly affects the speed of iterations.
Scan_capability_maximum_scan_angle	Defines the maximum scan angle to check for scan capability
Scan_capability_step_size	Check scanning capability in steps of this number starting from zero
Optimization_range	Defines the range that the array is trying to operate at with respect to targets
Max_beamwidth	Defines the maximum tolerable beamwidth for minimum sidelobe optimizations
Max_tolerable_sidelobe_level	Defines the maximum tolerable sidelobe level for minimum beamwidth optimizations
Edge_sidelobe_level	Defines the preferable sidelobe level at the pattern edges
Beam_direction_error_margin	Defines the tolerable error in beam direction
Penalty_invalid_edge_level	Defines the penalty for not satisfying the edge_sidelobe_level value
Penalty_invalid_sidelobe_level	Defines the penalty for not satisfying the maximum_tolerable_sidelobe_level
Penalty_invalid_beam_direction	Defines the penalty for not satisfying the Beam_direction_error_margin
Penalty_invalid_beamwidth	Defines the pnelty for not satisfying the max_tolerable_beamwidth
Penalty_grating_lobe	Defines the penalty for having grating lobes

When optimisation type is chosen to be fixed aperture size, the optimisation goal becomes minimising the sidelobe levels. Whereas if the type is chosen to be variable aperture size, the optimisation goal becomes minimising the beamwidth. But in genetic algorithm, a fitness function is used rather than a cost function. Since these values are inversely related, the fitness is computed differently for beamwidth optimisation and sidelobe level optimisation. The fitness function (optimisation goal) for beamwidth optimising genetic algorithm is solely defined as the current beamwidth, added with the penalties from feasibility checks. Note that, unlike random descent or simulated annealing no comparison is made to compute the fitness value.

$$delta = beamwidth_{current} + penalties \tag{5-7}$$

$$fitness = \frac{1}{delta} \tag{5-8}$$

The cost function for minimising beamwidth can be summarised as in 5-7, in which delta is tried to be minimised. Then based on this value a fitness value is calculated for an individual by taking the inverse of it as shown in 5-8. This value defines how a solution (individual) stands in a population.

It follows a similar logic when the optimisation type fixed aperture size. In this case, cost function (optimisation goal) for genetic algorithm is defined solely as the current sidelobe level, added with the penalties from feasibility checks.

$$delta = sidelobe_{current} + penalties \tag{5-9}$$

$$fitness = -delta \tag{5-10}$$

The cost function for minimising sidelobe level can be summarised as in 5-9, in which delta is tried to be minimised. Then based on this value a fitness value is calculated for an individual by taking the negative of it as shown in 5-10. The cost function for minimising beamwidth can be summarised as in 5 7, in which delta is tried to be minimised. Then based on this value a fitness value is calculated for an individual by taking the inverse of it as shown in 5 8. This value defines how a solution (individual) stands in a population. The way the fitness is computed is different for sidelobe level optimisation than the way it is for beamwidth optimisation mainly because beamwidth values are always positive, whereas sidelobe levels are negative. Regardless of how it’s computed, this value defines how a solution (individual) stands in a population.

An important thing to note is that any penalty high enough is going to immediately cause the fitness levels to drop to very low levels; effectively preventing them from being accepted into next population. Below presented parameters in Table 5-10 are chosen after a number of trials to yield the seemingly best results. Note that any text in italic in value column is a reference to a variable or a named constant also defined in the same table.

Table 5-10 Genetic algorithm parameters

Parameter Name	Possible Values/unit	Value
Optimization_type	Variable_aperture_size ; Fixed_aperture_size	<u>Not shared between different optimization types</u>
loadedPopulationSize	Positive integer	100
randomizedPopulationSize	Positive integer	100
numberOfVariables	Should be equal to total number of antenna elements	*NUM_TX+NUM_RX*
numberOfGenes	Should be an integer multitude of number of variables	*numberOfVariables**10
mutationProbability	A fraction between 0 and 1 inclusive. 0 for no mutations, 1 to mutate always	0.0625
tournamentSelectionParameter	A fraction between 0 and 1 inclusive. 0 for always choosing the	0.5

	worst individual, 1 for always choosing the best individual.	
numberOfGenerations	Positive integer	20
tournamentSize	Positive integer	10
numberOfReplications	Positive integer	2
Maximum_aperture_size	Positive real numbers in terms of wavelength	9
Scan_angles_degree	Vector of real numbers in degrees	[-89.9:0.1:89.9]
Scan_capability_maximum_scan_angle	Positive real number in degrees	30
Scan_capability_step_size	Positive real number in degrees	15
Optimization_range	Positive real number in terms of wavelength	3**Maximum_aperture_size*
Max_beamwidth	Positive real number in degrees	10
Max_tolerable_sidelobe_level	Negative real number in decibels	*Not shared between optimization types*
Edge_sidelobe_level	Negative real number in decibels	*Max_tolerable_sidelobe_level* – 1
Beam_direction_error_margin	Real positive number in degrees	0.5
Penalty_invalid_edge_level	Real number, multiplier; unitless	0

Penalty_invalid_sidelobe_level	Real number, multiplier; unitless	3
Penalty_invalid_beam_direction	Real number, multiplier; unitless	5
Penalty_invalid_beamwidth	Real number, multiplier; unitless	5
Penalty_grating_lobe	Real number, multiplier; unitless	3

Since this algorithm really is an analogy to biological evolution, the parameter choice was mostly done on a trial and error basis.

Fixed Aperture Size

For the sake of simplicity, the implementation of genetic algorithm doesn't differentiate between fixed antennas and movables. But chooses to ignore the two variables for transmitters on the edges. This way we can ensure a fixed aperture size. Its natural analogy is the unused genes in living organisms. The remaining elements (both the transmitters and the receivers) have the freedom to position between any point in between those elements. Below is the list of names and respective values for non-shared parameters for this type of optimization.

- **Optimization_type:** Fixed_aperture_size
- **Max_tolerable_sidelobe_level:** -9 dB

A maximum tolerable sidelobe level of -9 dB was chosen in contrast to -6 dB in previous algorithms, this is mainly due to disallow populations to have solutions with low sidelobe levels which are not tolerable in practice. Since this algorithm runs with random processes the results below are simply an example to what could be achieved. A summary of the results, plots of patterns at two different scan angles and a sketch of the array configuration can be found below. In Table 5-11 one of the array configurations obtained via running the random descent optimisation algorithms can be found, along with its computed sidelobe level and computed beamwidth.

Table 5-11 Sample optimised MIMO array with the use of random descent with fixed aperture size

	Value(s)	**Unit**
Transmit Antenna Positions	[0, 2.1378, 6.8534, 9]	Wavelength
Receive Antenna Positions	[0.3167, 3.4135, 3.9413, 4.5044]	Wavelength
Sidelobe Level	-10.4828	dB
Beamwidth	3.75	Degrees

In Figure 5-16 below, the array configuration's top view sketch can be found in detail. On top left the transmitter array positioning and on top right receiver array positioning can be seen separately. In addition, in the middle the superposed transmitter and receiver configuration the algorithm found originally can be observed. Finally, on bottom the virtual array element distribution can be observed.

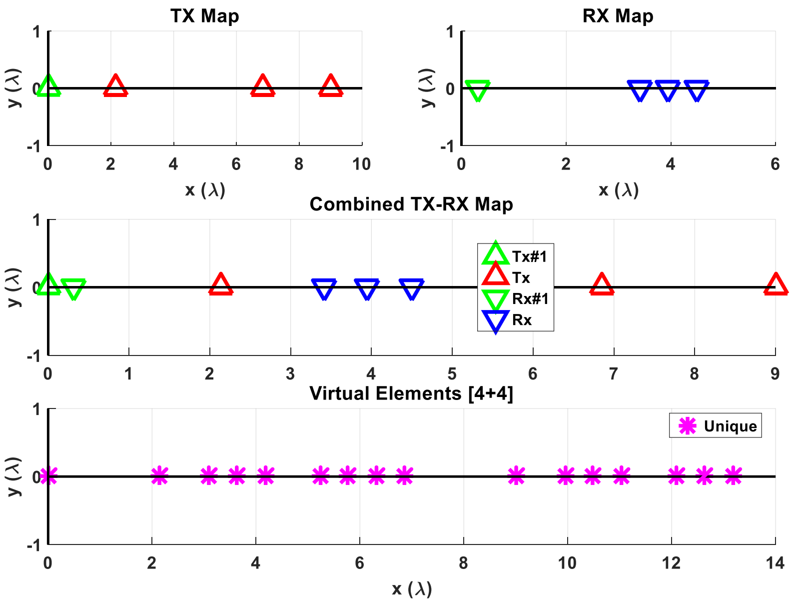


Figure 5-16 Exemplary simulated annealing fixed size array optimization configuration result

In Figure 5-17 the computed beam pattern can be seen, if such array was scanning at 0 degrees. The measured beamwidth is 3.75 degrees which is about 1.8 degrees less than a 5.56 degrees beamwidth expected from an array same size. And the sidelobe level is measured as -10.48 dB which is about 2.5 dB higher than the standard -13.1 sidelobe level.

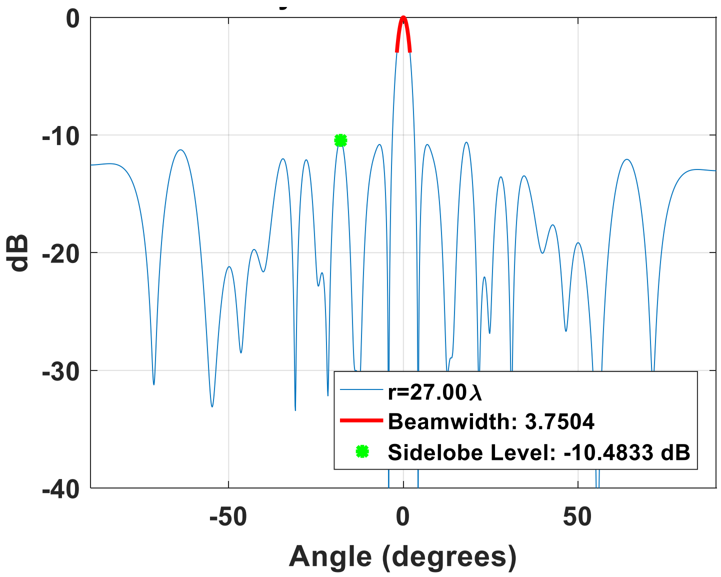


Figure 5-17 Exemplary genetic algorithm result scanning at zero degrees

In Figure 5-18 the computed beam pattern can be seen, if such array was scanning at 30 degrees. The measured beamwidth is 4.30 degrees which is about 2.12 degrees less than a 6.42 degrees beamwidth expected from an array same size. And the sidelobe level is measured as -9.32 dB which is about 4 dB higher than the standard -13.1 sidelobe level. The problem with this solution is however, that it has asymmetric scanning properties. As can be seen in Figure 5-19 not only the beam pattern when scanning to -30 degrees shows a maximum sidelobe level of -7.9 dB, which is higher than the predeclared maximum tolerable sidelobe level.

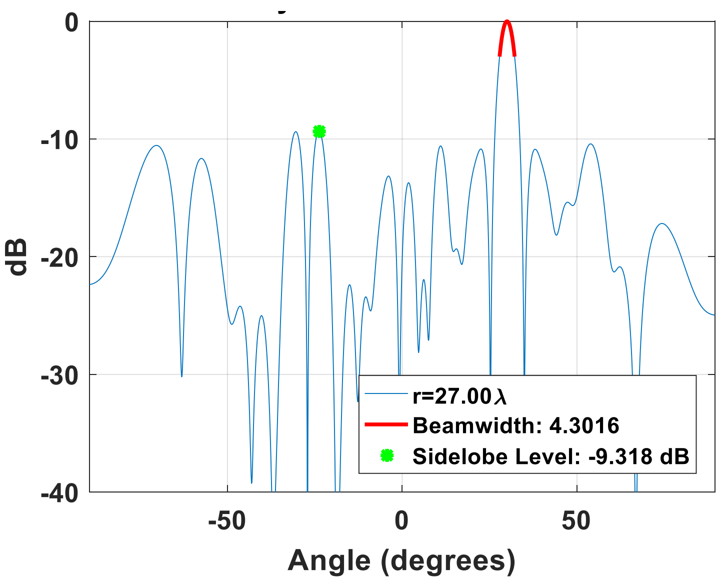


Figure 5-18 Exemplary genetic algorithm result scanning at 30 degrees

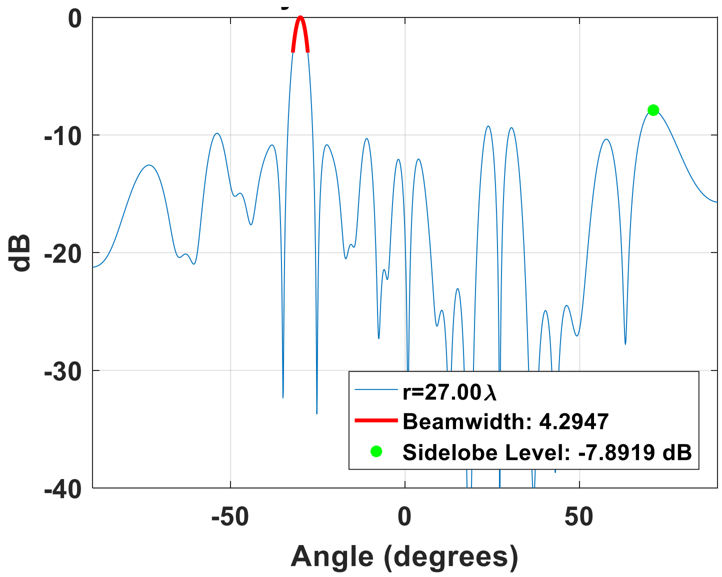


Figure 5-19 Exemplary genetic algorithm result scanning at -30 degrees

Variable Aperture Size

In variable aperture size optimization, one edge transmitter location is fixed to ensure varying physical aperture size with minimum redundant configurations with same aperture size. This is done again via ignoring the variable of the edge transmitter variable. However, the remaining elements (both the transmitters and receivers) have the freedom to take up any position in between those elements. Below is the list of names and respective values for non-shared parameters for this type of optimization.

- **Optimization_type:** Variable_aperture_size
- **Max_tolerable_sidelobe_level:** -9 dB

Contrary to the other optimisation type a higher maximum tolerable sidelobe level has been chosen, because this optimisation goal has proven itself to really push the aperture size to obtain narrow beamwidths which eventually causes the sidelobe levels to go down. Since this algorithm runs with random processes the results below are simply an example to what could be achieved. Below is an example configuration found via genetic algorithm with some beamwidth optimisation. A summary of this result, plots of patterns at two different scan angles and a sketch of the array configuration can be found below. In Table 5-12, the array configurations obtained via running the genetic algorithm can be found, along with its computed sidelobe level and computed beamwidth.

Table 5-12 Sample optimised MIMO array with the use of random descent with fixed aperture size

	Value(s)	**Unit**
Transmit Antenna Positions	[0, 2.4750, 4.8044, 6.5515]	Wavelength
Receive Antenna Positions	[0, 5.0959, 5.6294, 6.1147]	Wavelength
Sidelobe Level	-10.31	dB
Beamwidth	4.32	Degrees

In Figure 5-20 below, the array configuration's top view sketch can be found in detail. On top left the transmitter array positioning and on top right receiver array positioning can be seen separately. In addition, in the middle the superposed transmitter and receiver configuration the algorithm found originally can be observed. Finally, on bottom the virtual array element distribution can be observed.

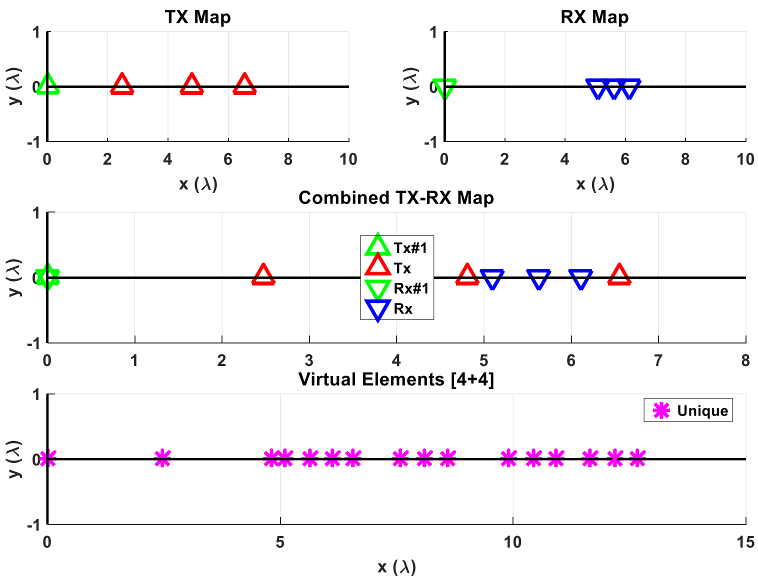


Figure 5-20 Exemplary genetic algorithm variable size array optimization configuration result

In Figure 5-21 the computed beam pattern can be seen, if such array was scanning at 0 degrees. The measured beamwidth is 4.32 degrees which is about 3.3 degrees less than a 7.63 degrees beamwidth expected from an array same size. And the sidelobe level is measured as -10.31 dB which is about 2.8 dB higher than the standard -13.1 sidelobe level. As before this configuration also shows asymmetric scanning properties. As can be seen in Figure 5-22 the beam pattern when scanning to -30 degrees shows a maximum sidelobe level of -9.2 dB, which is still lower than the predeclared maximum tolerable sidelobe level.

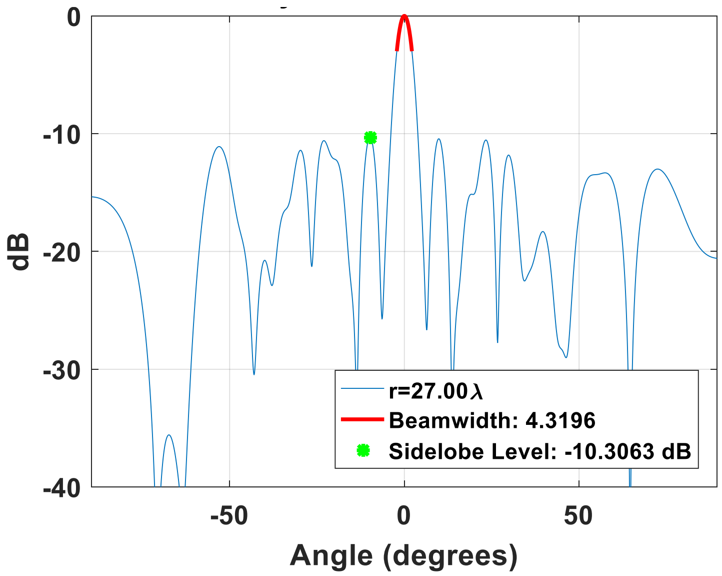


Figure 5-21 Exemplary genetic algorithm result scanning at zero degrees

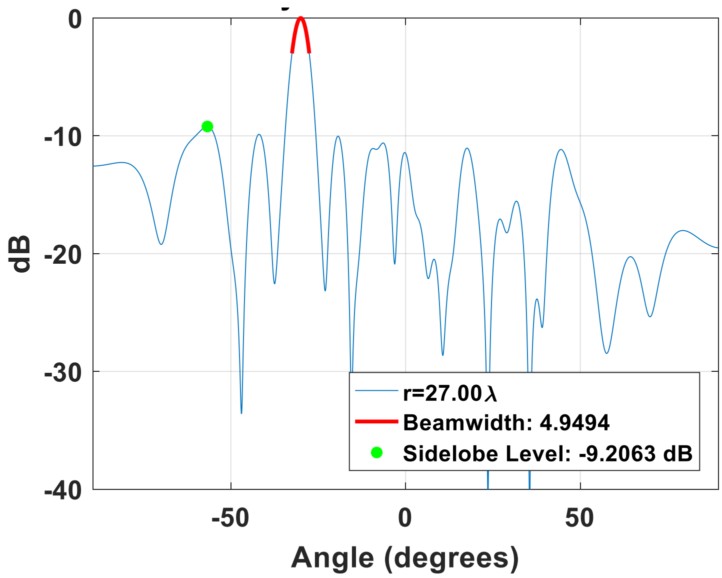


Figure 5-22 Exemplary genetic algorithm result scanning at -30 degrees

In Figure 5-23 the computed beam pattern can be seen, if such array was scanning at 30 degrees. The measured beamwidth is 4.98 degrees which is about 3.84 degrees less than a 8.82 degrees beamwidth expected from an array same size. And the sidelobe level is measured as -10.44 dB which is about 2.7 dB higher than the standard -13.1 sidelobe level.

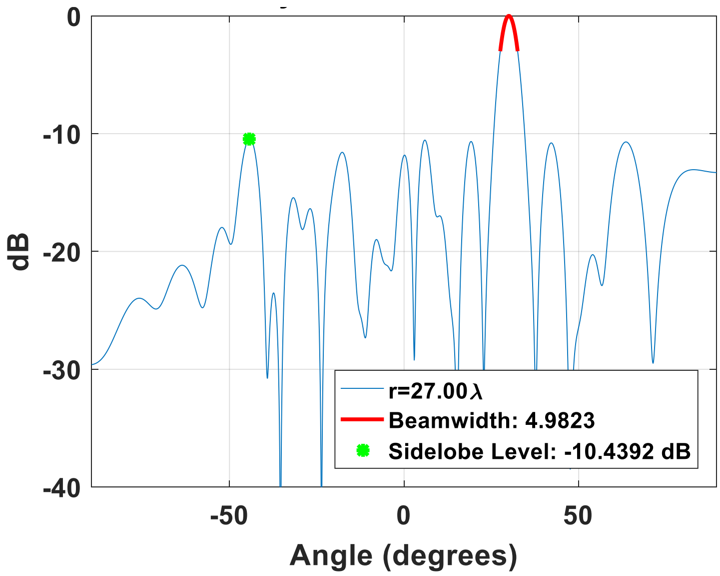


Figure 5-23 Exemplary simulated annealing algorithm result scanning at 30 degrees

This final configuration presented above was hand-picked among many other configurations including the ones presented before. This configuration showed the best improvement while still maintaining a feasible beam pattern at various angle and therefore it was then chosen to be used for experimental testing too.

5.4.4 Optimization Summary

We have shown that with the use of heuristic search algorithms, it is indeed possible to find configurations with trade-offs to optimize different goals such as minimizing the beamwidth or the sidelobes.

The results also show that given enough time, these optimization algorithms eventually reach local or close to global optima. So in order to get practical solutions, it might be useful to run these algorithms more than once and perhaps compare the results in a simulation or in an experimental medium.

Also, it can be observed that some of the optimal solutions are actually asymmetric configurations. And even though these configurations seem to offer symmetric beam patterns, asymmetric solutions do actually have a tendency to have asymmetric scanning characteristics even if they seem to offer symmetric patterns while scanning to 0 degrees. However, to overcome this issue constraints were imposed on the optimization and scanning on both sides were checked to ensure that asymmetry is tolerable (e.g. as long as the beamwidth and sidelobe levels are acceptable and the main lobe is where it is supposed to be, optimizers don't care about sidelobe structures).

Observing the evolution of the solutions also informed us the importance of constraint programming. In order for these algorithms to make smarter decisions, their constraints need to be programmed adequately (i.e. less time consuming for performance, better fitness functions).

The initial implementations of these search algorithms depend on randomizing the element locations. Future implementations can try to choose smarter element locations based on different constraints (e.g. proximity to other elements, a search history like in Tabu search algorithm[85][86]).

The results of these initial algorithms can be and were used to feed each other's initial solutions rather than sticking to a minor perturbation to the conventional MIMO configuration. This provided a more suitable optimization ground for genetic algorithm, but it could also make the algorithm approach to and get stuck in a local optimum, therefore it is better to use a hybrid population as the initial population.

5.5 Experimental Results

Upon reaching a configuration that was promising, a set of experiments were planned and conducted in the anechoic chamber to verify the optimisations experimentally. For this, the configuration that yielded the most optimisation while holding the initial requirements was used, which is the latest configuration obtained via genetic algorithm presented in this chapter. To recap, the configuration of the MIMO array is as in Table 5-13. And the optimization parameters that were used to reach this result were to try and increase the beamwidth starting from a conventional MIMO array while trying to keep -9dB sidelobe levels. This number was chosen because a 3-4 dB loss in sidelobe levels were assumed to be tolerable for practical applications. It was also found through trial and error that asking for anything more than -9dB was simply blocking the way of significant improvements.

Table 5-13 Configuration and theoretical performance of the optimised MIMO array for experimental testing

	Value(s)	**Unit**
Transmit Antenna Positions	[0, 2.4750, 4.8044, 6.5515]	Wavelength

Receive Antenna Positions	[0, 5.0959, 5.6294, 6.1147]	Wavelength
Sidelobe Level	-10.31	dB
Beamwidth	4.32	Degrees

Compared to other results obtained this configuration yields the most optimised beamwidth while still offering a <-9dB sidelobe level even while scanning at obscure angles. This was especially important when it was compared to the configuration presented at Table 5-11 which should –in theory- yield a 3.75 degree beamwidth but displays asymmetric scanning capabilities which has lower than optimal sidelobe levels when scanning to -30 degrees.

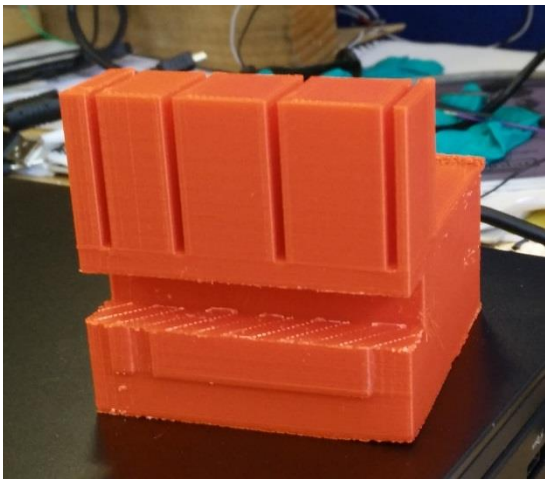


Figure 5-24 Photo of the aperiodic MIMO sensor array housing without the sensor modules

When the sensor modules were placed, the rest of the system was contained in a single box. This kept the technology demonstrator still modular and portable. A photo of the system with the data acquisition box can be seen in Figure 5-25.

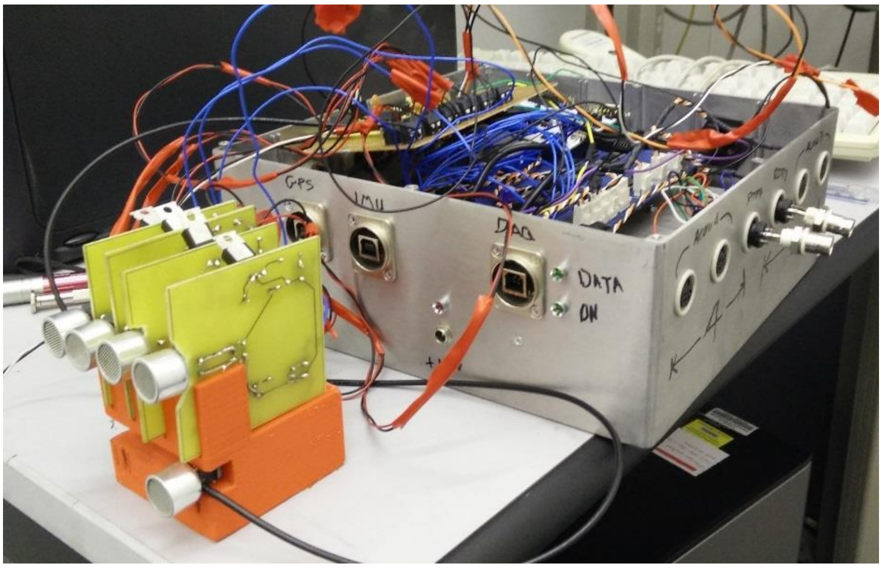


Figure 5-25 Photo of the aperiodic MIMO sensor array housing with the sensor modules and the data acquisition box

The photos of the experimental setups can be found in the next section explaining the experiment scenarios. Below in Table 4-2 again is the summary of the shared parameters of the experiments.

Table 5-14 Experiment parameters

Experiment Property	**Value**	**Unit**
Number of Tx	4	-
Number of Rx	4	-
Carrier Medium	Acoustic	-
Carrier Frequency	40	kHz
Wavelength	8.375	mm
Waveform	Up-chirp LFM	-
Bandwidth	4	kHz
Range Window	Gaussian	-
Windowed Range Resolution	0.15	m
Angular Window	None	-
Tx Sensor Beamwidth	55	Degrees
Tx Sensor Bandwidth	2	kHz
Rx Sensor Beamwidth	55	Degrees
Rx Sensor Bandwidth	2	kHz

Receive Gain	~20	dB
Tx Element Positions	[0, 20.73, 40.24, 54.87]	mm
Rx Element Positions	[0, 42.68, 47.15, 51.21]	mm
Fraunhofer Range	>2.5	m
Sample Rate	250	kilosamples/s
Pulse length	0.08	sec
PRI	0.25	sec
Number of Pulses	3	-

As mentioned in the previous chapter, experimental setup was not ideal. Some of the setup's shortcomings were summarised as;

- non-coherent sampling of the analogue data due to acquisition equipment's characteristics,
- imperfect anechoic chamber,
- reflections from other background elements (equipment itself, chamber door etc.).

Out of these problems the most critical was the non-coherent sampling of the analogue data due to acquisition equipment's characteristics. But these problems were solved with the same methods explained and applied in previous chapter.

5.5.1 CLEAN Method for Post-Processing

In radar and/or sonar systems, the reflection image from a single scatterer is usually characterized as the point spread function. In our specific case the results are obtained in 2-dimensional range-azimuth space. The point spread functions usually contain sidelobes even after suppression methods like array tapering and waveform windowing. In such cases, a range-azimuth map obtained from an area that contains multiple close-by scatterers with different radar cross sectional areas would yield a highly complex map. In such a map, weaker targets might disguise as sidelobes of stronger targets. Or the reverse situation can happen; sidelobes of multiple targets can constructively add-up to create spikes in the map that might be mistaken as a scattering centre. In order to overcome these problems and to make the analysis of signal returns a cleaning algorithm called CLEAN was developed which is mainly used in radio astronomy [87].

CLEAN Background

Before applying the CLEAN algorithm directly, a literature survey has been done to find similar approaches to similar or same problems. [88] is an analysis of CLEAN that shows that it is a statistically correct fitting procedure, along with an analysis of under which conditions it's applicable to use CLEAN which is beneficial for us to see if it is indeed applicable in our case. [89] explains the mathematical background of the method and discusses the ambiguity of CLEAN algorithm. And finally it explains a new approach for the processing of maps with limited phase information. [90] performs CLEAN first to improve MUSIC performance because MUSIC algorithm needs to an input parameter of the number of targets to function properly. [91] proposes a method to create maximum-entropy maps. They propose a randomly producing maps whose transforms may or may not fit the measured signals (data). After discarding the ones that do not fit at all, and keep producing maps and they come up with the same maps, many times. They take the one that is produced the most times, and they call it the maximum-entropy map, also explained as the map that is most likely to represent the reality. [92] demonstrates an application of CLEAN technique in radio-astronomy with the use of large random thinned arrays that provide high angular resolution, which is quite a similar application to ours.

CLEAN Method

CLEAN algorithm is a fairly simple image processing algorithm with minimal adjustable parameters and minimal amount of required guess-work (compared to MUSIC for example [93]) which was first published by Jan Högbom. Due to variations and modifications on the original method, the original method is also often referred to as the Hogbom-CLEAN. In order to get maximum performance CLEAN algorithm was tried with absolute and complex values (Coherent CLEAN), fitted and expected range patterns and method was established with methods yielding the best results regarding majority of all experiments.

CLEAN algorithm works on any dimensional input data, however since our results contain 2-dimensional maps that will be the focus. After the application of beamformers a range-azimuth map of a ground scan is obtained. CLEAN algorithm begins after an image is produced, which is generally called the *dirty map*. First, the strongest peak in the whole map is found and its strength and coordinates are recorded. Then, a same-size empty map is filled with the expected beam pattern of a point scatterer in that coordinates of the map (point spread function); this map is called the *dirty beam*. The next step is simply subtraction of the *dirty beam* from the *dirty map*. This subtraction can occur either with absolute values (magnitude) or complex values. In ideal conditions, one would expect complete removal of a target and its sidelobes but, non-idealities in beamforming, noise and interference conditions and for other reasons this is not always the case. Also, if complex values are used for subtraction then there arises a risk of phase mismatch. A phase mismatch during the subtraction can actually cause the map to become *dirtier*. The algorithm then proceeds to deal with the next strongest peak in the map to be removed. The next strongest peak is also processed the same way, and so on. This iteration stops until a pre-set condition is met. This condition is usually either a maximum number of iterations or minimum strongest peak level.

Because our implementation of the algorithm works with phase information and deals with complex data, an extra “entropy cannot increase” condition was introduced. This means, after the subtraction step, maximum entropy (strongest peak value) cannot be higher than the previous one. If somehow this condition is met, it is assumed that map does not contain any more coherent targets (at least worthy of processing). The process until this point is usually summarized as the *deconvolution* part of the CLEAN. A flowchart of this algorithm can be seen in Figure 5-26.

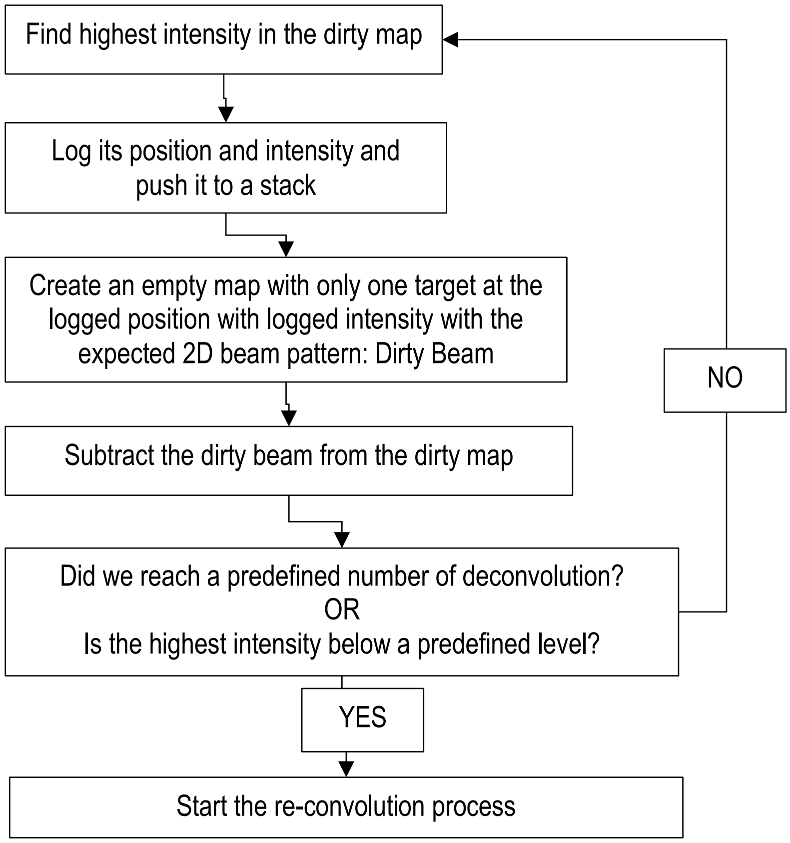
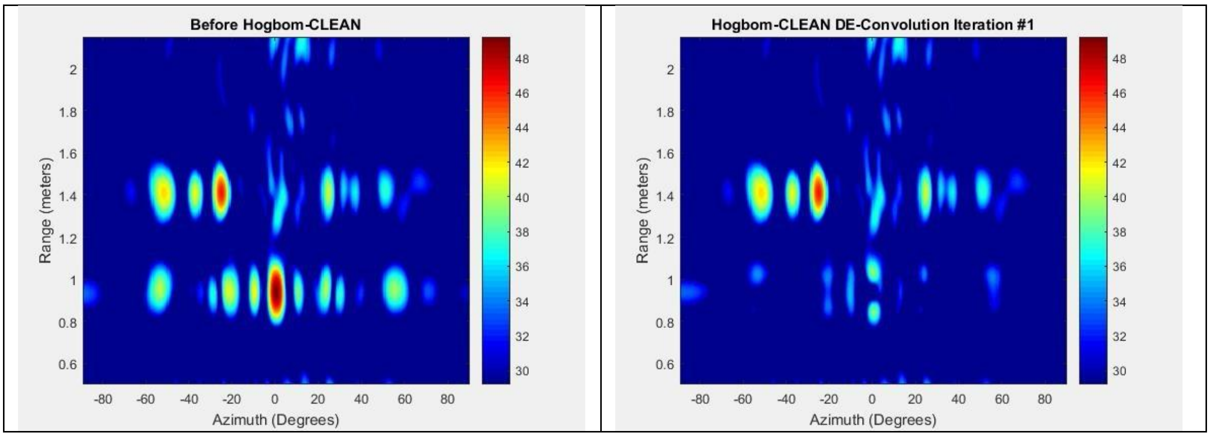


Figure 5-26 Flowchart explaining the deconvolution part of CLEAN

In addition, the process of deconvolution can be observed in Figure 5-27. Note that, for the sake of simplicity iterations #3-18 are skipped as they focus on relatively smaller reflections.



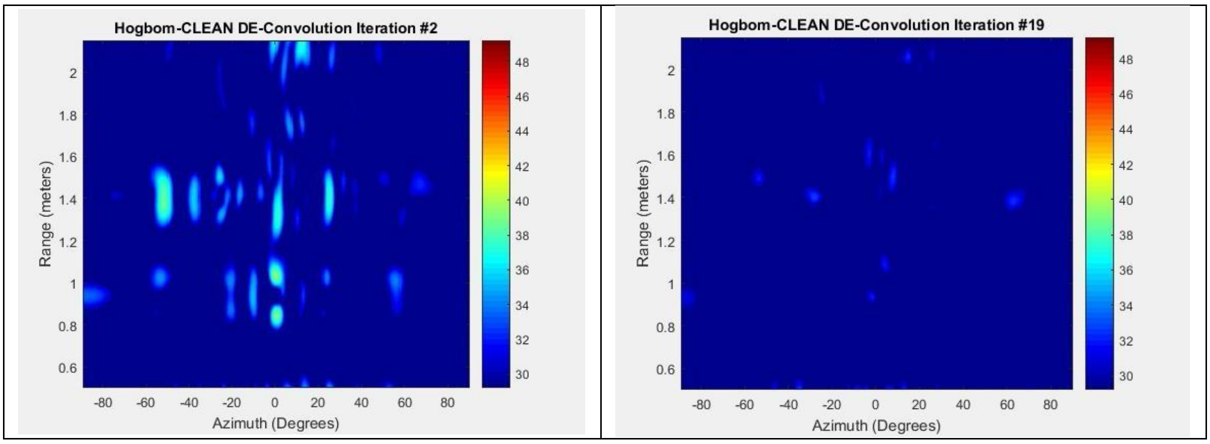


Figure 5-27 CLEAN deconvolution process example steps

Next part of the CLEAN is the *re-convolution*. Using the recorded strength and coordinate data, a cleaned version of the *dirty beams* are super-imposed onto the output of the *deconvolved* map. This addition can be done on an empty map too, but often output of the deconvolved map is used to preserve remaining information. This cleaned version of *dirty beam* -often called *clean beam*- is usually a Gaussian return having the same shape of the main lobe of the dirty beam and nothing else. The *clean beam*'s strength is usually multiplied with a variable called loop gain, which is a value from 0 (exclusive) to 1. In our implementation loop gain is set to 1, because there was no apparent reason to reduce the target strengths when they are added back to the map. Also worth noting is that the order of addition is unimportant, because the *re-convolved* map is simply a superposition of *clean beams* and the remaining background data. A flowchart of this algorithm can be seen in Figure 5-28.

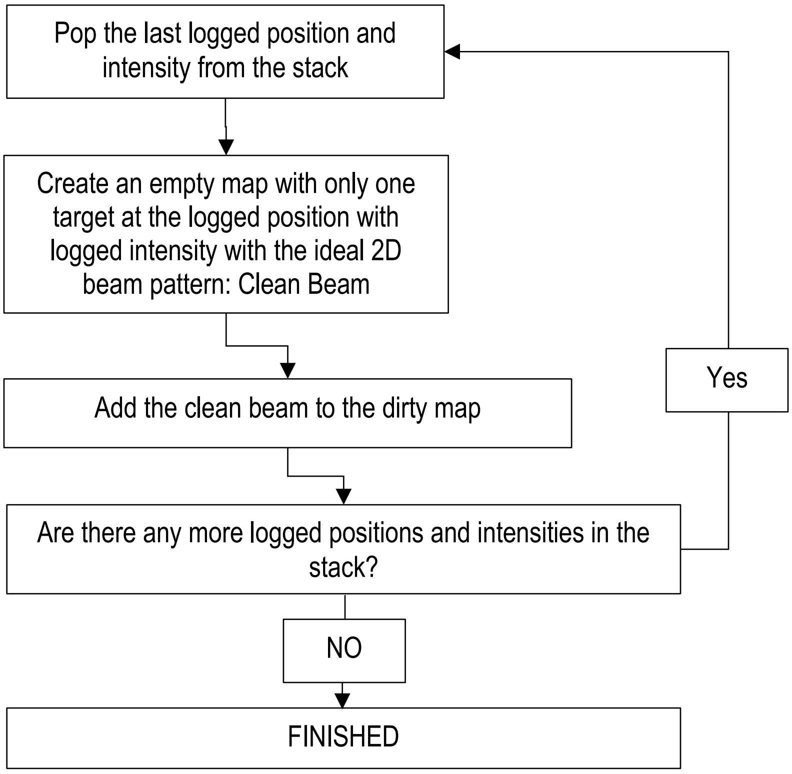


Figure 5-28 Flowchart explaining the deconvolution part of CLEAN

In addition, the process of reconvolution can be observed in Figure 5-28. Note that, for the sake of simplicity iterations #3-18 are skipped as they focus on relatively smaller reflections.

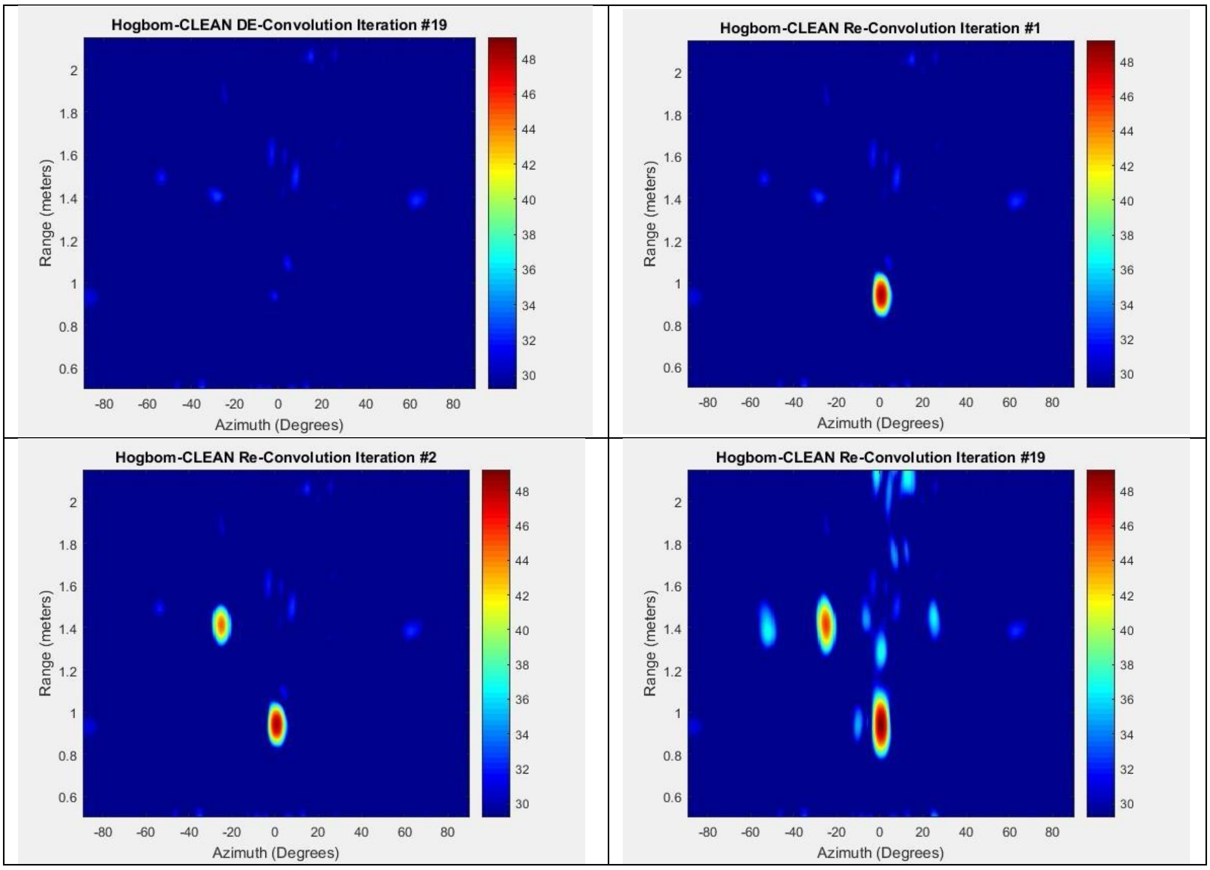


Figure 5-29 CLEAN deconvolution process example steps

In order for CLEAN to work properly, a good computation of the *dirty beam* is required. If a proper *dirty beam* cannot be computed, then the subtraction will leave artefacts on the map. On the other hand, if the *dirty beam* is well computed, CLEAN should yield only the main lobe in the map. So the performance of CLEAN is dependent on 1) how well the expected beam pattern can be computed, and 2) how well the beamforming results will match the expected beam pattern. So in a way, performance of CLEAN algorithm can actually be used as a metric for our technology demonstrator's performance. If the CLEANed results show single target(s) without (or highly reduced) sidelobes, that means our expectations of beam patterns match the experimental results and vice versa.

Initial Experimental and Simulated Results with CLEAN

After the development of the algorithm, testing and verification of the algorithm was done both via simulations and experiments using the new aperiodic MIMO setup (see Table 4-2) also using near-field algorithms. A scenario with 2 targets was setup, so that the targets wouldn't fall into each other's sidelobe patterns, but the cross-section of the sidelobe patterns would add-up to present a stronger return at an unexpected coordinate. A second scenario was to place a weaker target at another stronger target's sidelobe regions and to use CLEAN to isolate only the targets. Second scenario was tested only with simulations

Scenario 1

First scenario involves two different targets at different range and different azimuth bins. The expected outcome of this simulation is that the range and azimuth sidelobes of these targets would constructively add up along with ambient noise to create strong return points. The sketch of the scenario can be found in Figure 5-30.

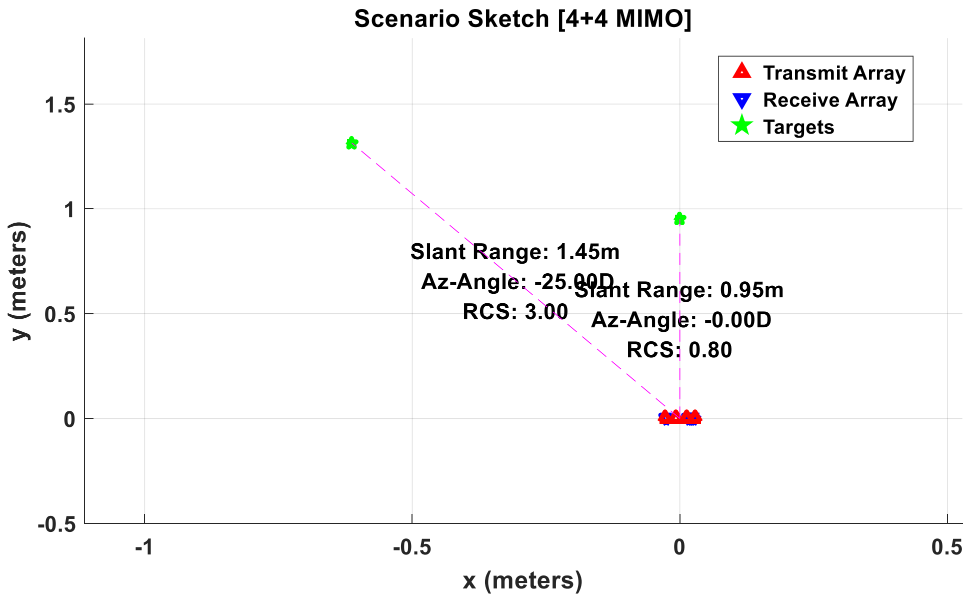


Figure 5-30 Sketch of the scenario 1; two targets at different range and different azimuth bins

Properties of the targets can be found in Table 5-15. Parameters of the targets has been set to match experimental values. In this scenario it is expected to have two intersections of azimuth and range sidelobes of the targets. One residing at around -25 degrees azimuth and 0.95 metres range and another residing at around 0 degrees azimuth and 1.45 metres range. These sidelobes were expected to add up constructively or destructively.

Table 5-15 Properties of targets in CLEAN testing scenario 1

	Range	**Azimuth**	**Reflectivity**
Target 1	0.95 metres	0 degrees	0.8 m^2
Target 2	1.45 metres	-25 degrees	3 m^2

In Figure 5-31 the range-azimuth map obtained via simulations without CLEAN of the 1st scenario can be seen. In the simulation results, it can be observed that at around -20 degrees azimuth, 0.95 metres range target sidelobes add up constructively almost imitating multiple small scatterers. Whereas we don't observe constructive addition of sidelobes at 0 degrees azimuth and 1.45 metres range. This phenomenon can be simply explained by the difference of phase returns of the signals from the targets, but it's out of the scope of this section. Instead isolating actual targets from "fake" targets will be proceeded with.

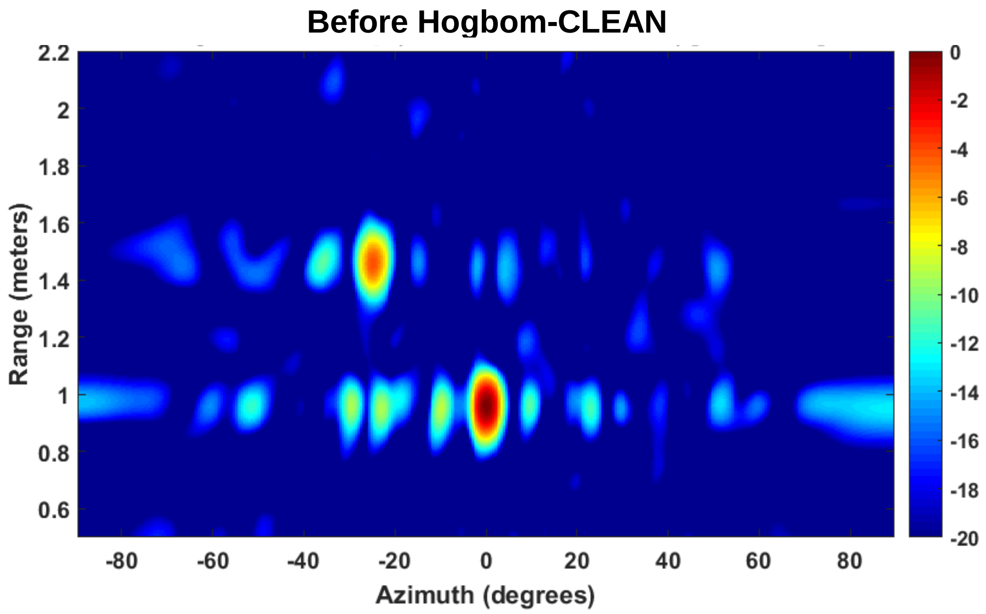


Figure 5-31 Raw range-azimuth map of simulations of the 1st CLEAN testing scenario

In Figure 5-32 the range-azimuth map obtained via simulations with CLEAN of the 1st scenario can be seen. When CLEAN algorithm is applied, it coherently removes the effects of the sidelobes. Both constructively and destructively added up sidelobes are removed from the range-azimuth map. It's a rewinding process of coherent superposition of targets. Therefore, ideally it leaves no traces except background noise in the range-azimuth map. When the results of the CLEAN algorithm are inspected for first scenario, it can be observed that not only all sidelobes are removed but also no artefacts due to their additions remain as well. And that only the actual 2 targets remain in the map. When the previously specified area is inspected (around -20 degrees azimuth, 0.95 metres range) no trace of sidelobes or any strong returns apart from those at the target locations can be seen anymore.

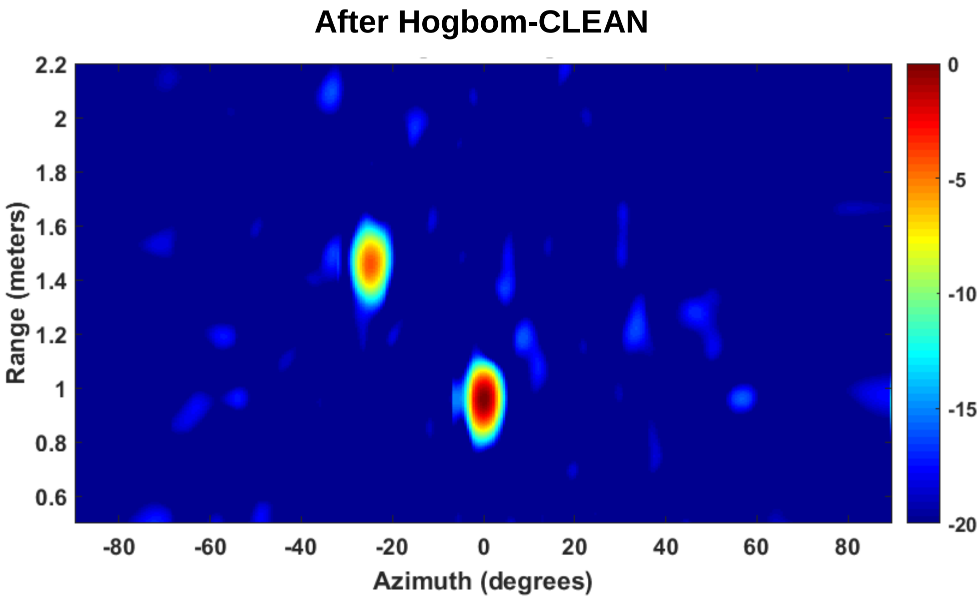


Figure 5-32 CLEANed range-azimuth map of simulations of the 1st CLEAN testing scenario

Below in Figure 5-33 is the range-azimuth map obtained via experiments without CLEAN of the 1st scenario can be seen where a physical setup matching the scenario sketch was used. The experiments were done in an anechoic chamber with the Aperiodic MIMO Sonar. When Figure 5-33 and Figure 5-31 is compared, the higher relative sidelobe levels (about 2-3 dB) than in simulation can be observed due to non-ideality of real world, but the locations of the sidelobes are the same. However, this gives a better chance to test out the CLEAN algorithm. At 1.45 metres range and -55, -40 and 25 degrees azimuth multiple relatively strong reflections can be observed. Other relatively strong reflections can also be observed at 0.95 metres range and -55, -20, -10, 25 and 55 degrees azimuth. These returns have relative reflection strengths around -8 dB. In a practical situation, this level could be too much to separate real targets from fake targets or sidelobes.

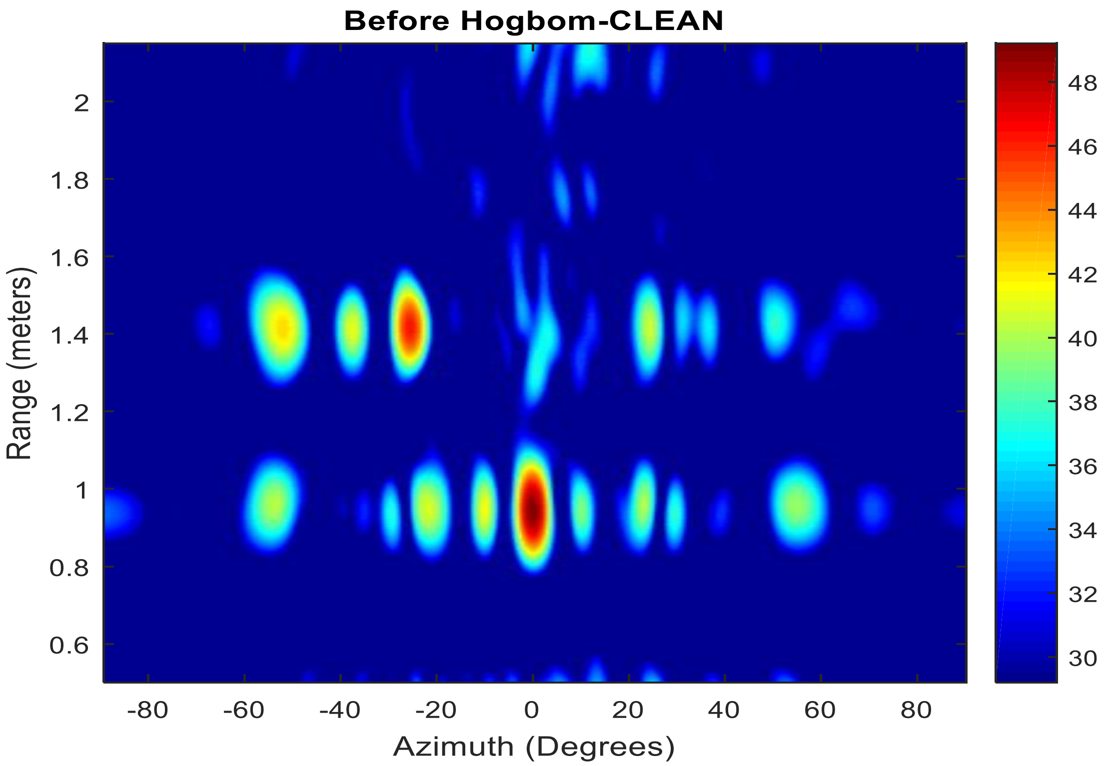


Figure 5-33 Raw range-azimuth map of experiments with the 1st CLEAN testing scenario

Below in Figure 5-34 is the range-azimuth map obtained via experiments with CLEAN of the 1st scenario can be seen. Like in the simulated results, we observe that almost all sidelobes are removed from the map or decimated to relative levels of -14dB. The CLEANed map clearly shows where real targets lie. Since this is an experimental setup the obtained patterns do not completely match the ideal patterns (i.e. mismatch between the experimental *dirty-beam* and the *constructed beam*). Due to this reason the *deconvolution* process is also imperfect. Therefore, sidelobe reduction levels similar to simulations are not reachable. But it is possible to optimise the process by better modelling the dirty beam as mentioned before.

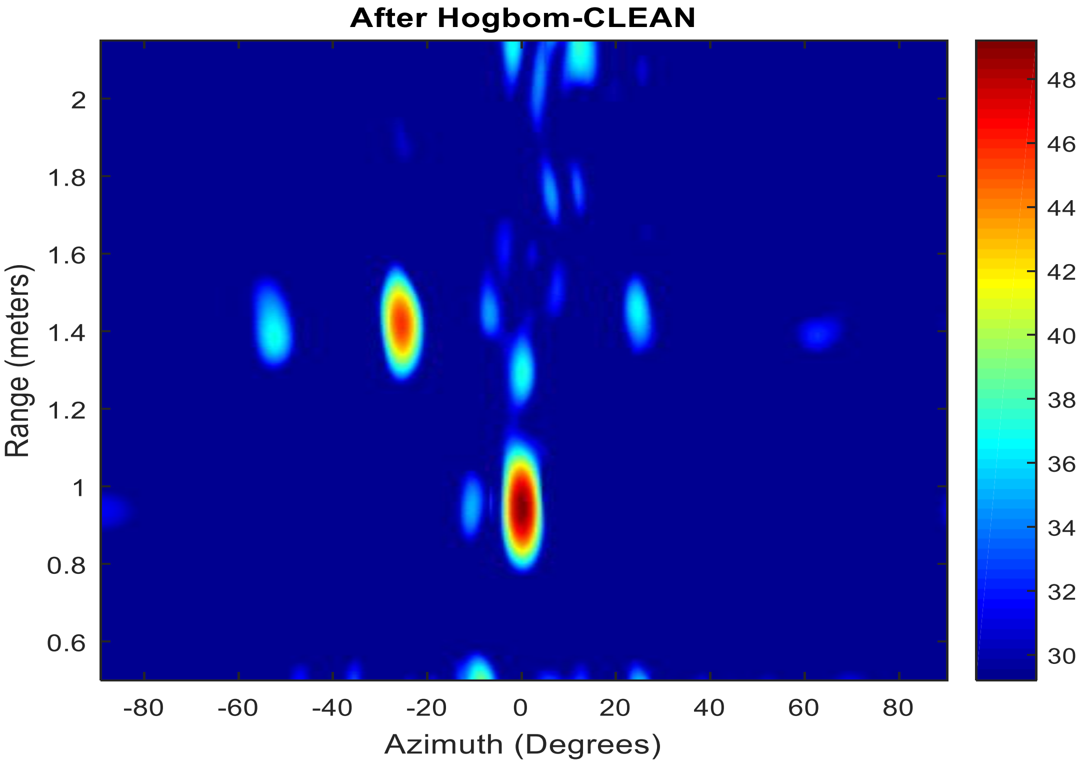


Figure 5-34 CLEANed range-azimuth map of experiments with the 1st CLEAN testing scenario

<u>*Scenario 2*</u>

Second scenario involves 3 targets at different range and different azimuth bins. One of the targets is weaker than the others and is placed at an intersection point of other targets' sidelobes, so that it would possibly dissolve or disguise as a sidelobe in the range-azimuth map. The sketch of the scenario can be seen in Figure 5-35.

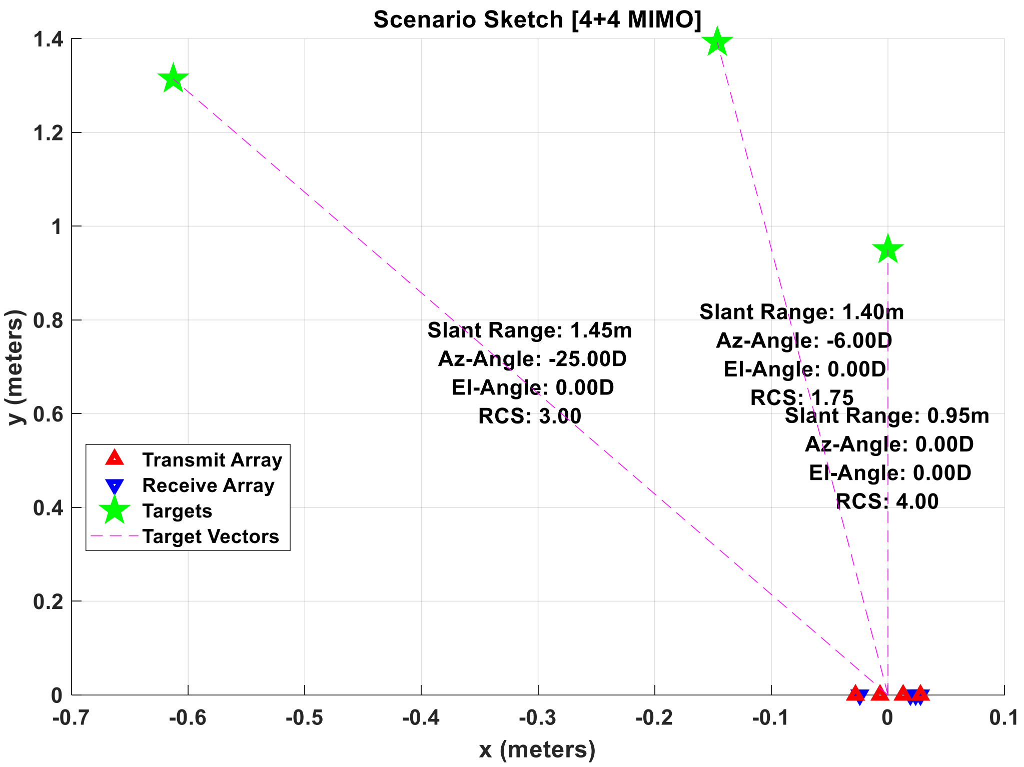


Figure 5-35 Sketch of the scenario 2; 3 targets at different range and different azimuth bins, with one of them residing at the intersection of others' sidelobes

Properties of the targets can be found in Table 5-16. Parameters of the first 2 targets has been set to match experimental values and properties of the third target has been set to yield us results suitable to test CLEAN algorithm (i.e. low RCS). In this scenario it is expected to have two intersections of azimuth and range sidelobes of the targets. One residing at around -25 degrees azimuth and 0.95 metres range and another residing at around 0 degrees azimuth and 1.45 metres range. Another weaker target was also added at -6 degrees azimuth and 1.4 metres, expecting it to be disguised next to these intersections. These intersections and the new target are expected to add up constructively or destructively.

Table 5-16 Properties of targets in CLEAN testing scenario 2

	Range	**Azimuth**	**RCS**
Target 1	0.95 metres	0 degrees	4 m^2
Target 2	1.45 metres	-25 degrees	3 m^2

Target 3	1.40 metres	-6 degrees	1.75 m^2

In Figure 5-36 the range-azimuth map obtained via simulations without CLEAN of the 2nd scenario can be seen. In the simulation results, it can be observed that at around -20 degrees azimuth, 0.95 metres range target sidelobes add up constructively almost imitating multiple small scatterers (like the previous results). In addition, different from the previous results relatively weak -but similar to other intersection-scatterers at around -5 degrees and 1.4 metres can be observed. When the map is observed, it's not very clear to point out the third target. It is lost between the sidelobes of other targets.

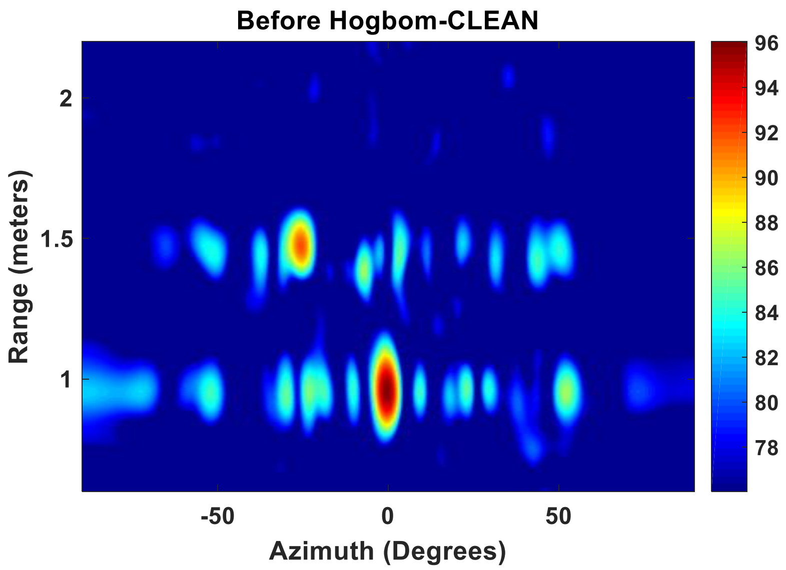


Figure 5-36 Raw range-azimuth map of simulations of the 2nd CLEAN testing scenario

In

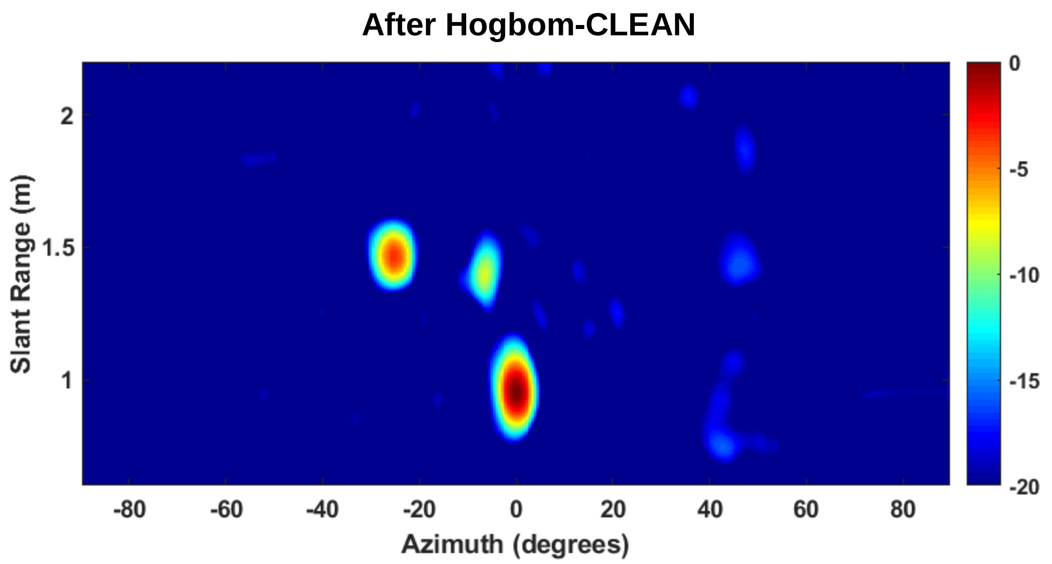


Figure 5-39 the range-azimuth map obtained via simulations with CLEAN of the 2nd scenario can be seen. When the results of the CLEAN algorithm for this scenario is inspected, it can be observed that not only all sidelobes are removed like in the first scenario but also that the third target is also isolated from the sidelobes and now can be seen clearly. When the previously specified area (around -6 degrees azimuth, 1.4 metres range) is inspected it can be seen that strong reflections of intersection are gone and only the target signature is left. Below in Figure 5-37 the deconvolution steps for this dateset can be observed. Furthermore in Figure 5-38 reconvolution steps for this dataset can be seen. Note that iteration steps #4-8 are skipped for the sake of simplicity.

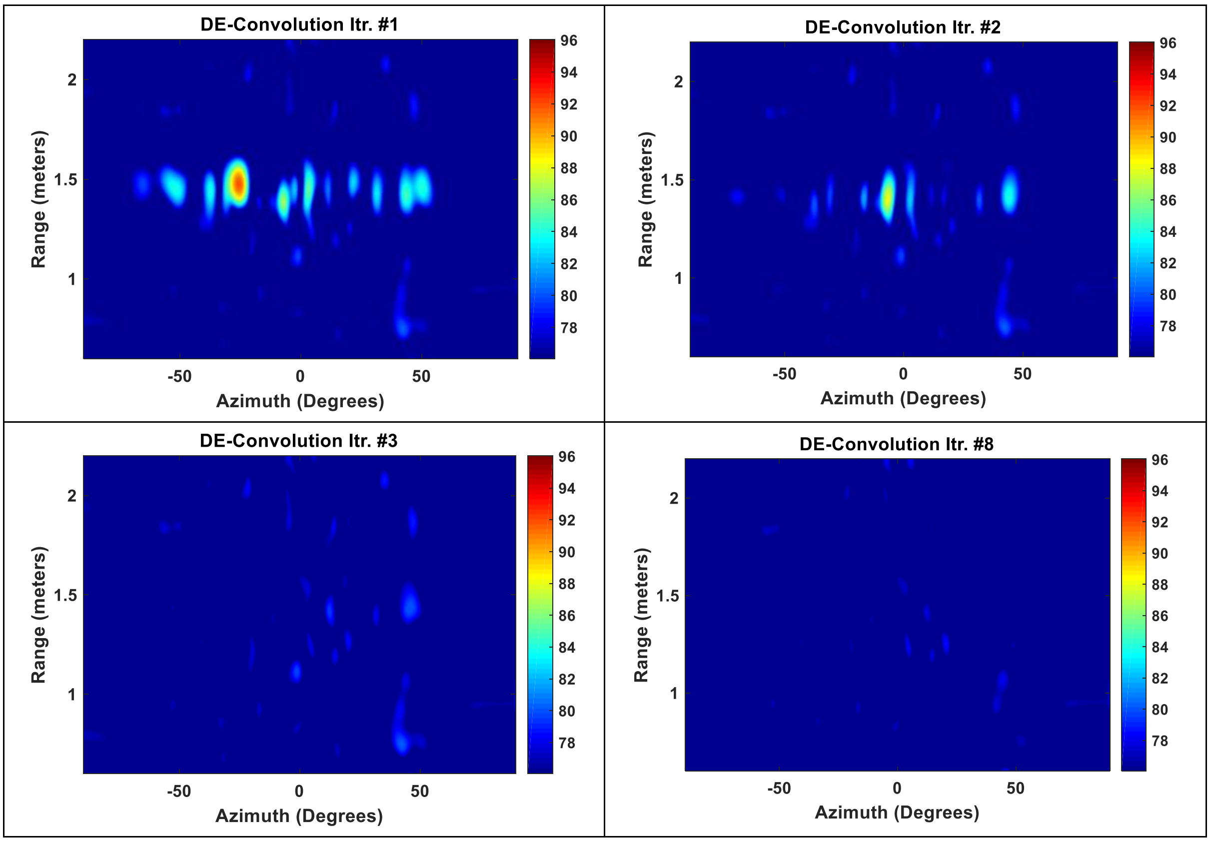
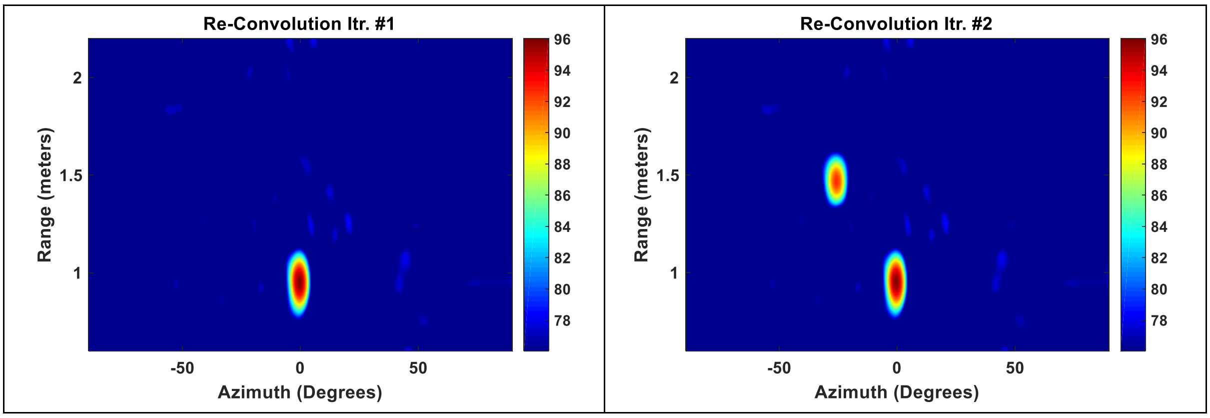


Figure 5-37 Deconvolution steps of a scenario where a less reflective target is hidden in sidelobes



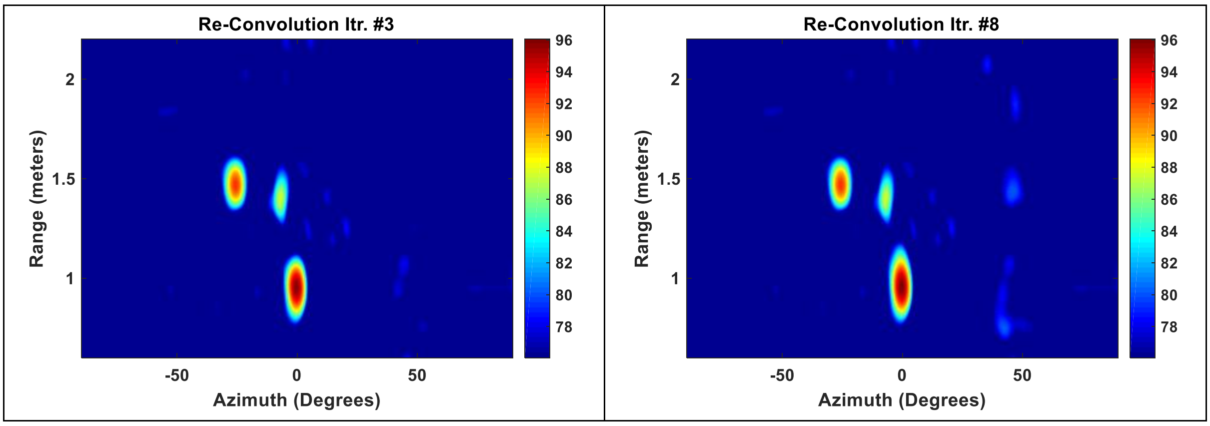


Figure 5-38 Reconvolution steps of a scenario where a less reflective target is hidden in sidelobes

Since CLEAN algorithm is a post-processing method we modified and used to clarify our results, which is the testing and verification of Aperiodic MIMO, further experimental testing of CLEAN hasn't been done. However, where applicable CLEAN algorithm has been used to clarify and support our findings in the rest of this chapter.

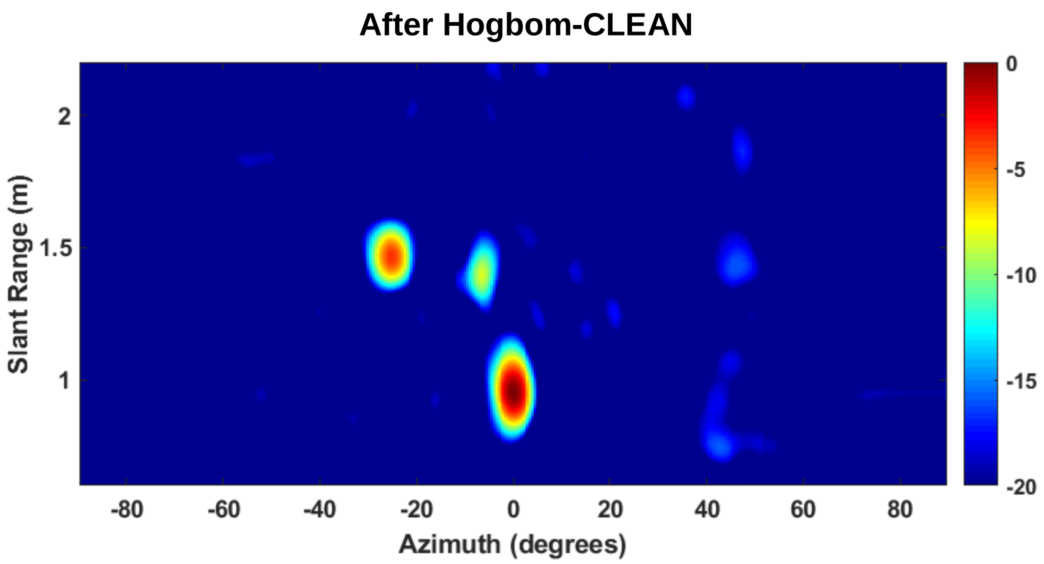


Figure 5-39 CLEANed range-azimuth map of simulations of the 1st CLEAN testing scenario

The results presented above prove that simulations match the experimental results, and that the effect of the CLEAN algorithm is the same for both types of results. And since the simulations match the experiments, CLEAN algorithm can therefore be used with confidence for presentation of further results and can even be used as a verification tool to prove that the expected (simulated) sidelobe structures match experimental results.

5.5.2 Scenarios and Results

The aperiodic MIMO housing was built and tested in the anechoic chamber. Various experiments were performed where target(s) were placed about 1.8m away from the sonar and changing azimuth angles. Target measurements were made with 0, -5, -15, 15, 20 and 25 degrees to test the angular width over which beamforming can operate. These scenarios are designed to check beamforming and target localization capabilities, azimuth resolution. After the experiments with calibrated targets, a mountain bicycle was put inside the anechoic chamber and sonar measurements were made to grab an image of the bike. This experiment was to demonstrate the capabilities of the technology demonstrator for an extended target. The experimental scenarios are given with photos of the physical setup inside anechoic chamber and with their corresponding scenario sketches. After each scenario presentation, the results obtained with and without CLEANing are presented.

Experiments with Calibrated Targets

First, experiments were done with calibrated targets as before to verify the computed parameters such as beamwidth and sidelobe levels. These experiments were performed where target(s) were placed at an average of 1.8m away from the sonar and changing azimuth angles.

Target at 0 Degrees

In Figure 5-40 the over the top sketch of the first scenario can be seen. A target is placed at about 2.2m slant range with about 0 degrees angle. The photo of the setup can be seen in Figure 5-41. This scenario was designed to verify single target detection capabilities at near zero scan angle (beamwidth, sidelobe levels, etc.).

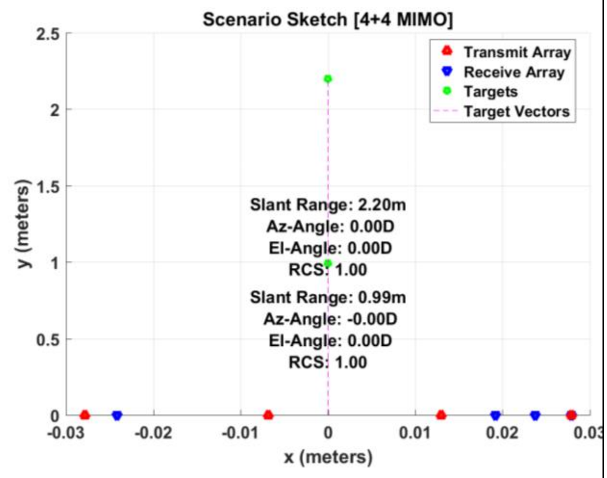


Figure 5-40 Target at 0 degrees; scenario sketch in Cartesian coordinate system

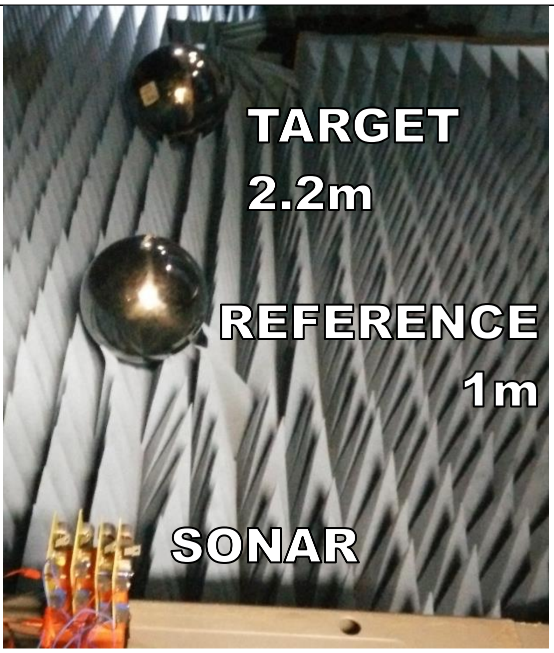


Figure 5-41 Target at 0 degrees; photo of the experimental setup

In Figure 5-42 the range-azimuth map obtained from measurements with the omission of the phase calibration reference and without any post-processing method. A strong reflection at about 0 degrees and 2.2 meters can be observed. This location coincides with the target's actual location.

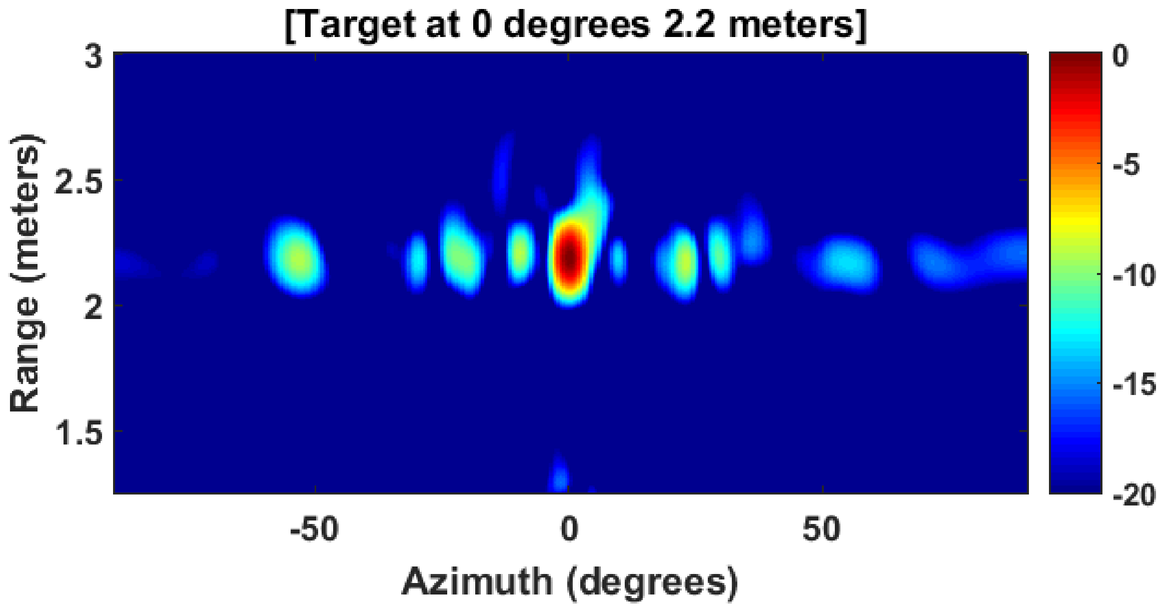


Figure 5-42 Target at 0 degrees; range-angle map (not CLEANed)

In Figure 5-43 the range-azimuth map obtained from measurements with the omission of the phase calibration reference and with CLEAN post-processing. A strong reflection at about 0 degrees and 2.2 meters can be observed. This location coincides with the target's actual location. Apart from that we see that almost all sidelobes and artefacts from the background are gone, leaving only the return in the middle as the sole target signature. As mentioned before in CLEAN section, this is a good indicator of how well the experimental beam patterns match the expected beam patterns. Note that CLEAN algorithm works by estimating the beam patterns based on the array structure and not based on the experimental results.

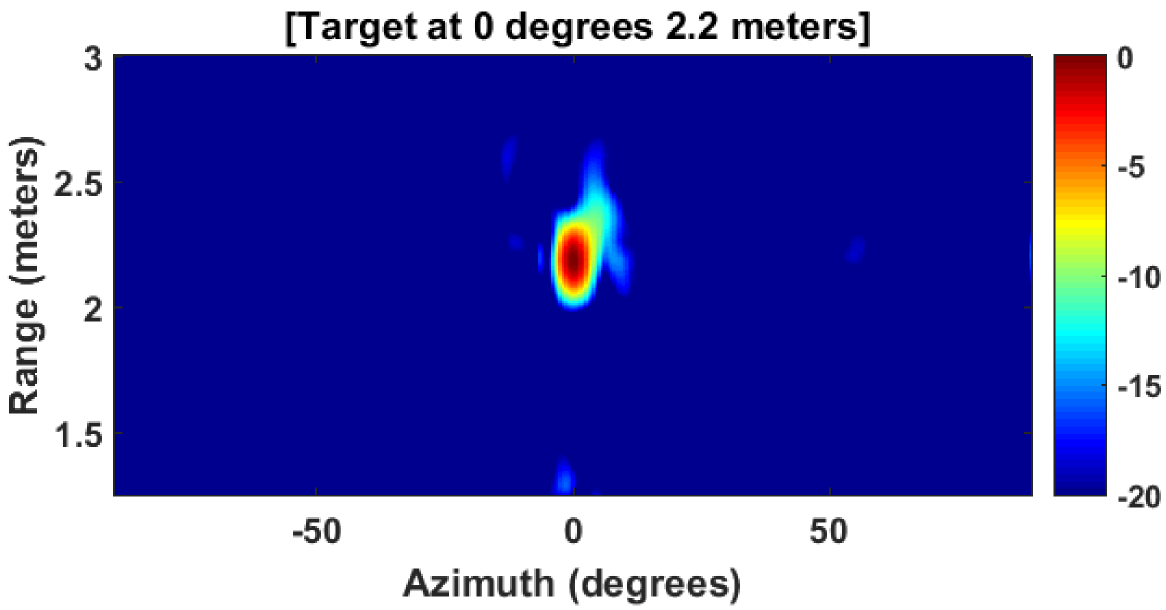


Figure 5-43 Target at 0 degrees; range-angle map (CLEANed)

In Figure 5-44 the azimuth cut from the range-angle map at the target range of 2.18 meters can be seen. In the angle cut beamwidth is measured is about 4.25 degrees, which coincides with the simulated beamwidth of 4.27 degrees. Also in the angle cut the sidelobe level can be seen as -8.8dB which is less than expected compared to simulated level of -10.22 dB, however this can be explained due to noise and other imperfections in the sensor positioning, non-ideality of anechoic chamber and reflections from the reference (i.e. multi-path interference, range and angle sidelobes adding up etc.).

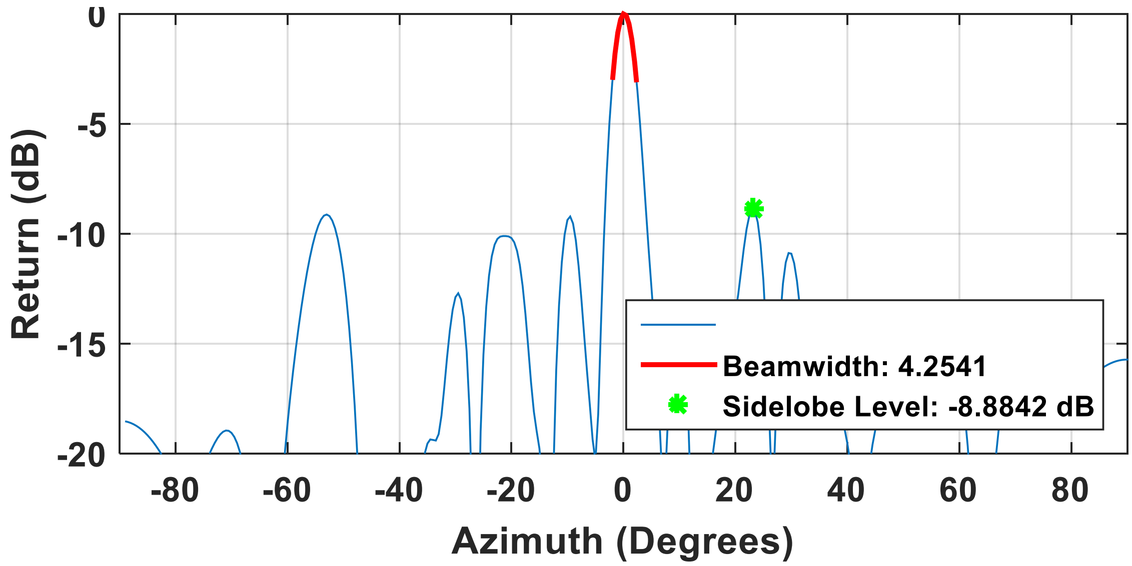


Figure 5-44 Target at 0 degrees; azimuth cut at target range (not CLEANed)

<u>*Target at -5 Degrees*</u>

In Figure 5-45 the over the top sketch of the second scenario can be seen. A target is placed at about 1.9m slant range with about -5 degrees angle. The photo of the setup can be seen in Figure 5-46. This scenario was designed to verify single target detection capabilities (beamwidth, sidelobe levels, etc.) at a small angle to test scanning symmetry.

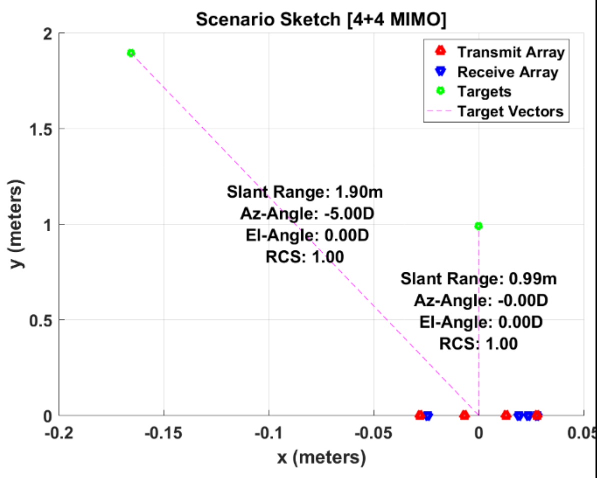


Figure 5-45 Target at -5 degrees; scenario sketch in Cartesian coordinate system

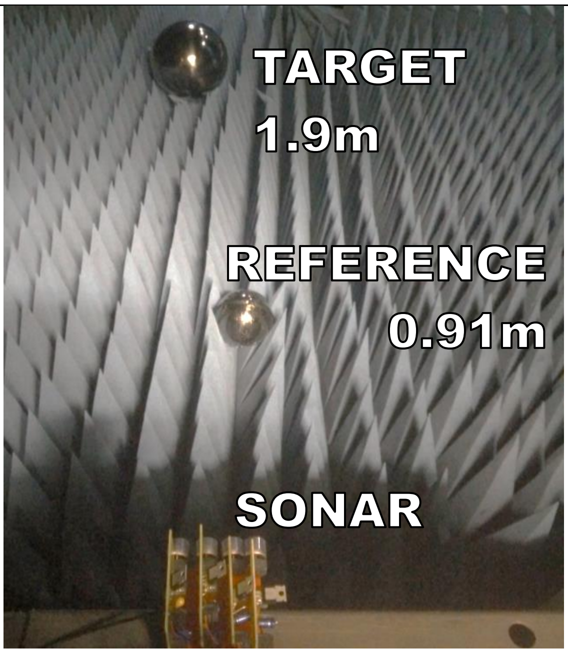


Figure 5-46 Target at -5 degrees; photo of the experimental setup

In Figure 5-47 the range-azimuth map obtained from measurements with the omission of the phase calibration reference and without any post-processing method. A strong reflection at about -5 degrees and 1.9 meters can be observed. This location coincides with the target's actual location.

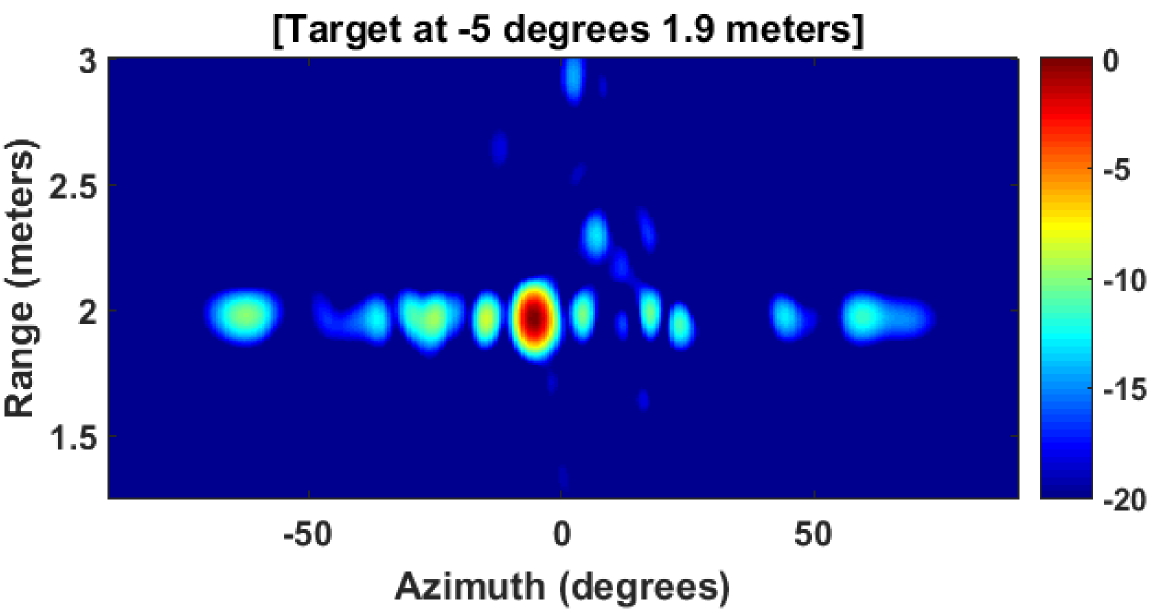


Figure 5-47 Target at -5 degrees; range-angle map (not CLEANed)

In Figure 5-48 the range-azimuth map obtained from measurements with the omission of the phase calibration reference and with CLEAN post-processing. A strong reflection at about -5 degrees and 1.9 meters can be observed. This location coincides with the target's actual location. Apart from that we see that almost all sidelobes and artefacts from the background are gone, leaving only the return in the middle as the sole target signature. As mentioned before in CLEAN section, this is a good indicator of how well the experimental beam patterns match the expected beam patterns.

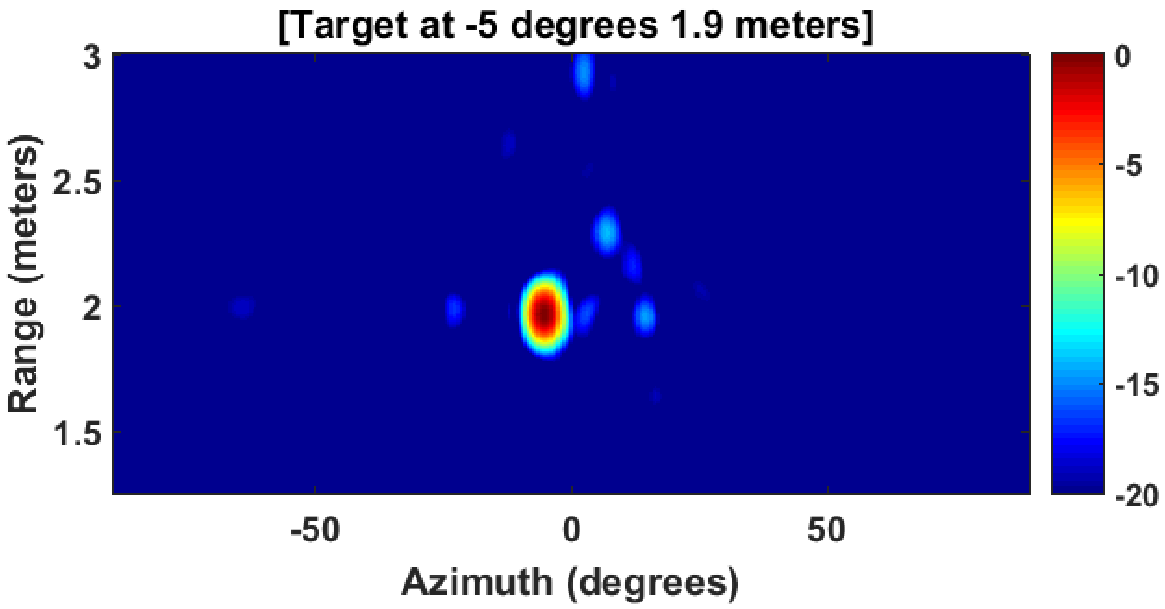


Figure 5-48 Target at -5 degrees; range-angle map (CLEANed)

In Figure 5-49 the azimuth cut from the range-angle map at the target range of 1.96 meters can be seen. In the angle cut beamwidth is measured is about 4.25 degrees, which coincides with the simulated beamwidth of 4.29 degrees. Also in the angle cut the sidelobe level can be seen as -8.5dB which is less than expected compared to simulated level of -10.16 dB, however this can be explained due to noise and other imperfections in the sensor positioning, non-ideality of anechoic chamber and reflections from the reference (i.e. multi-path interference, range and angle sidelobes adding up etc.).

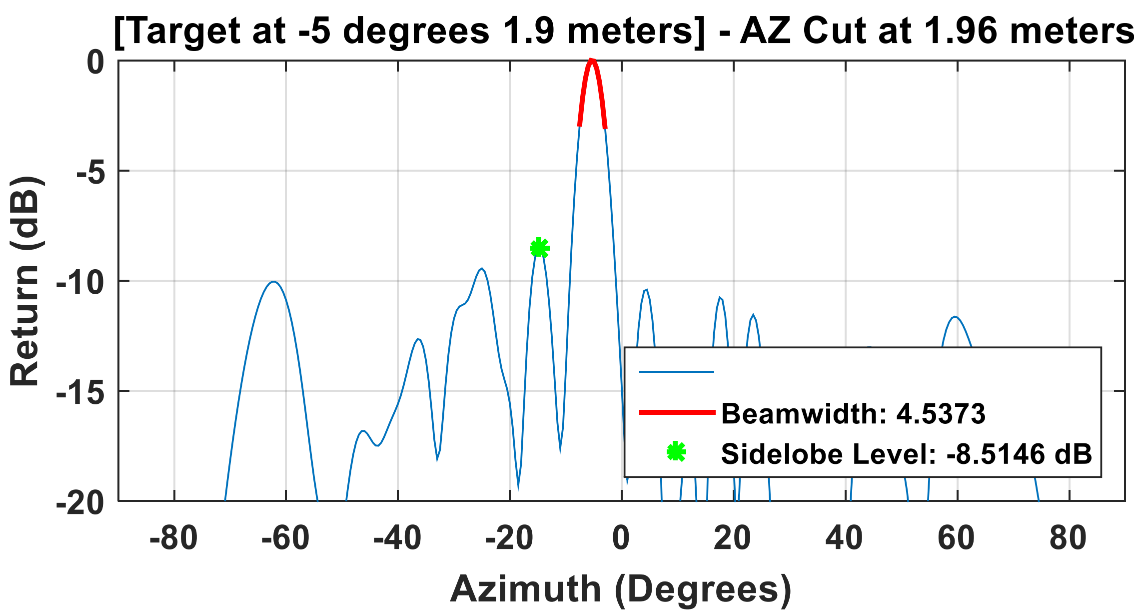


Figure 5-49 Target at 0 degrees; azimuth cut at target range (not CLEANed)

<u>*Target at -15 Degrees*</u>

In Figure 5-50 the over the top sketch of the second scenario can be seen. A target is placed at about 1.4m slant range with about -15 degrees angle. The photo of the setup can be seen in Figure 5-51.

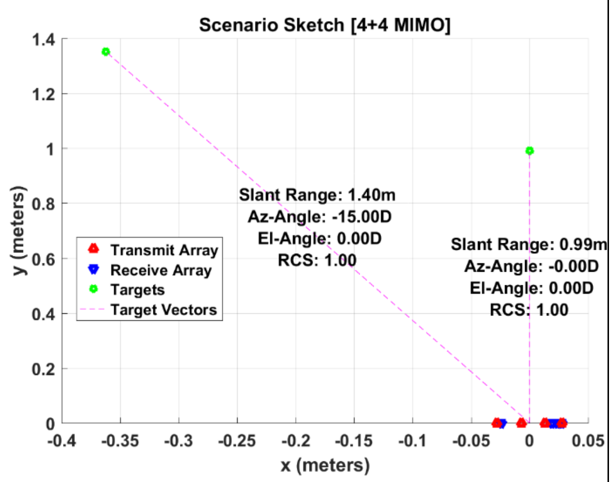


Figure 5-50 Target at -15 degrees; scenario sketch in Cartesian coordinate system

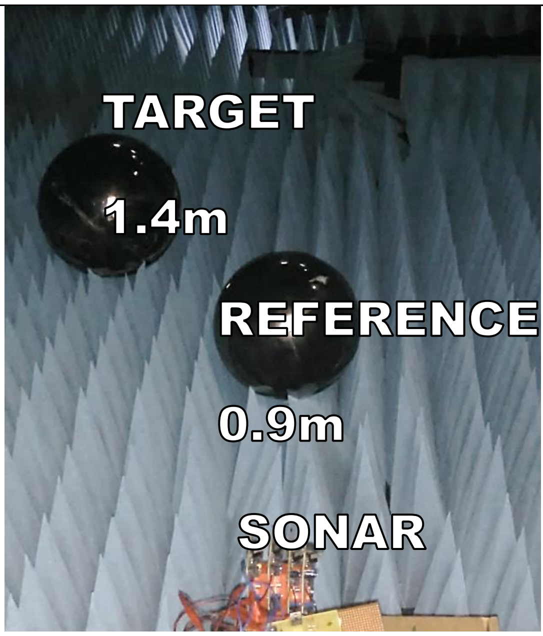


Figure 5-51 Target at -15 degrees; photo of the experimental setup

In Figure 5-52 the range-azimuth map obtained from measurements with the omission of the phase calibration reference and without any post-processing method. A strong reflection at about -15 degrees and 1.4 meters can be observed. This location coincides with the target's actual location.

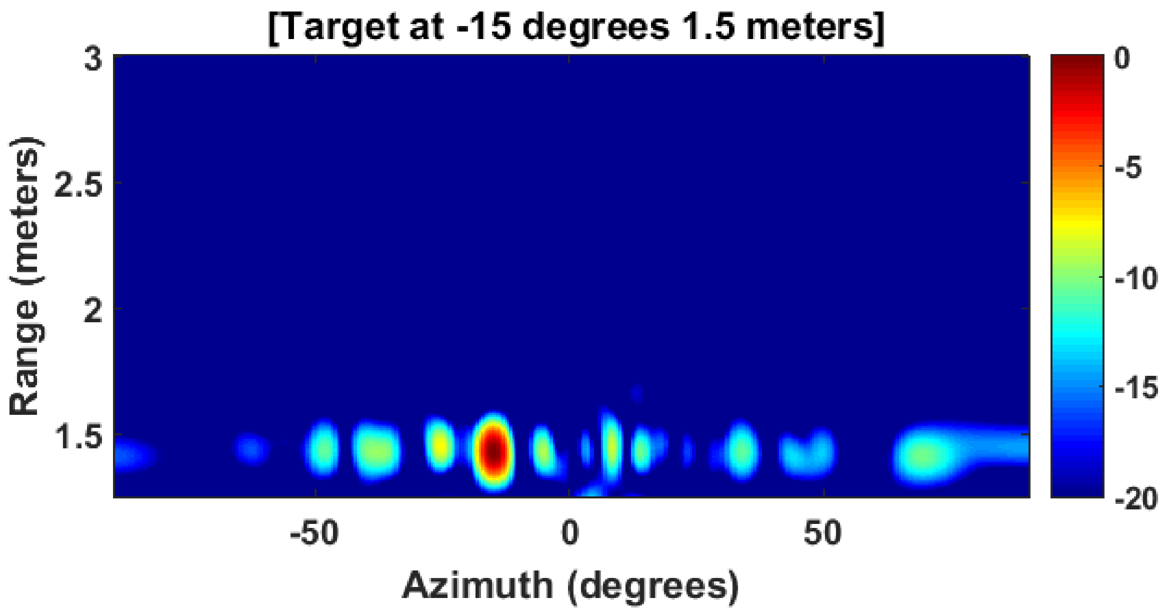


Figure 5-52 Target at 15 degrees; range-angle map (not CLEANed)

In Figure 5-53 the range-azimuth map obtained from measurements with the omission of the phase calibration reference and with CLEAN post-processing. A strong reflection at about -15 degrees and 1.4 meters can be observed. This location coincides with the target's actual location. Apart from that we see that almost all sidelobes and artefacts from the background are gone, leaving only the return in the middle as the sole target signature. As mentioned before in CLEAN section, this is a good indicator of how well the experimental beam patterns match the expected beam patterns.

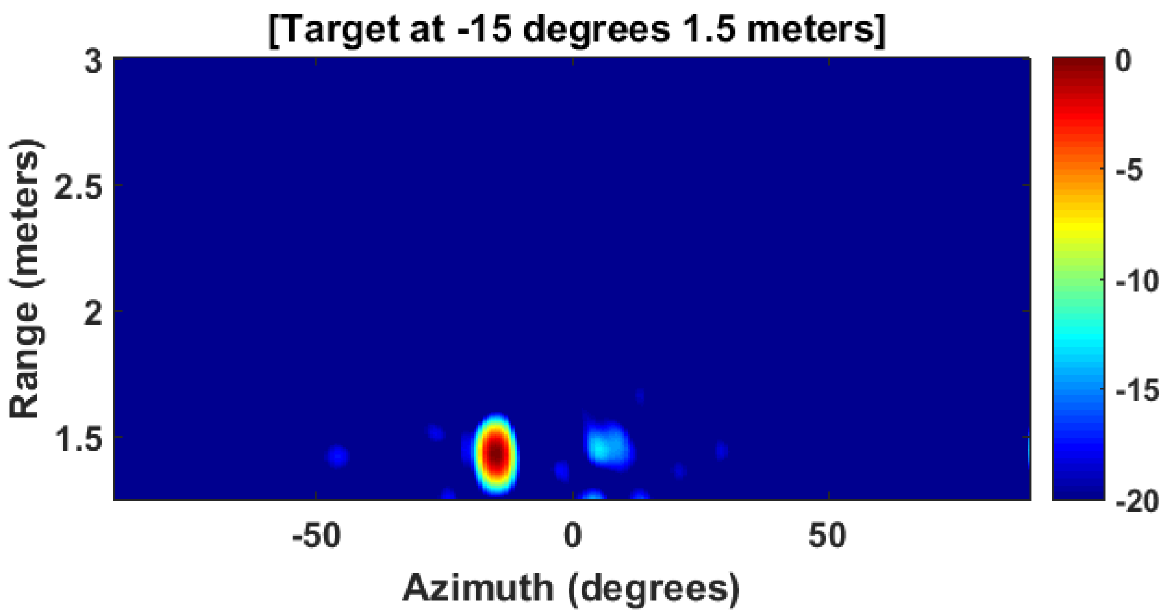


Figure 5-53 Target at 15 degrees; range-angle map (CLEANed)

In Figure 5-54 the azimuth cut from the range-angle map at the target range of 1.43 meters can be seen. In the angle cut beamwidth is measured is about 4.11 degrees, which is in the 10% vicinity of the simulated beamwidth of 4.42 degrees. Also in the angle cut the sidelobe level can be seen as -7.94 dB which is less than expected compared to simulated level of -10.06 dB, however this can be explained due to noise and other imperfections in the sensor positioning, non-ideality of anechoic chamber and reflections from the reference (i.e. multi-path interference, range and angle sidelobes adding up etc.).

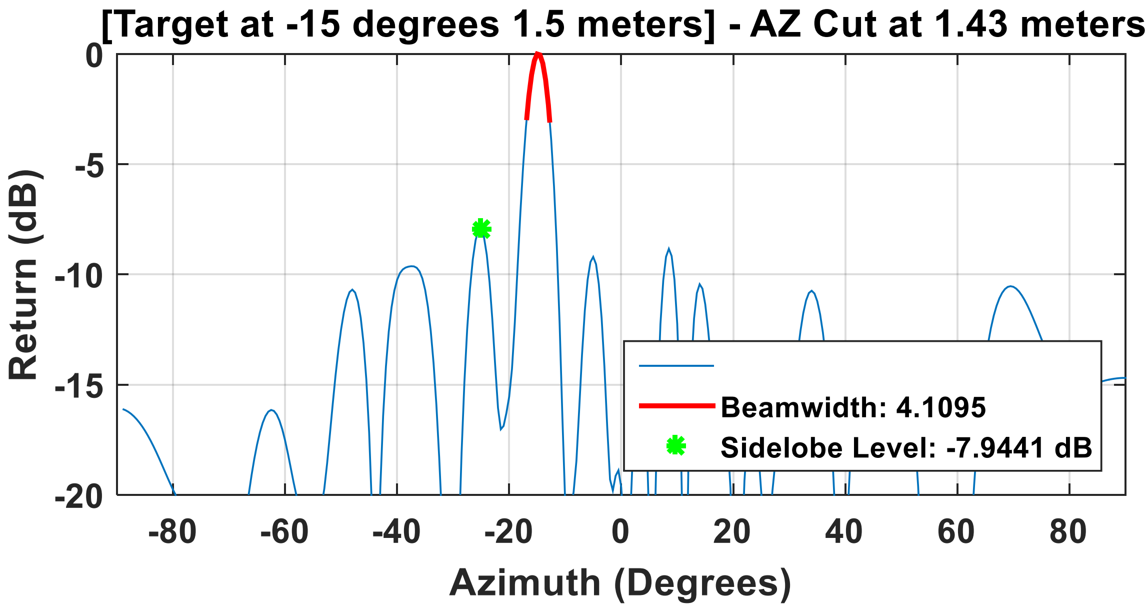


Figure 5-54 Target at 15 degrees; azimuth cut at target range (not CLEANed)

<u>*Target at 15 Degrees*</u>

In Figure 5-55 the over the top sketch of the second scenario can be seen. A target is placed at about 1.8m slant range with about 15 degrees angle. The photo of the setup can be seen in Figure 5-56.

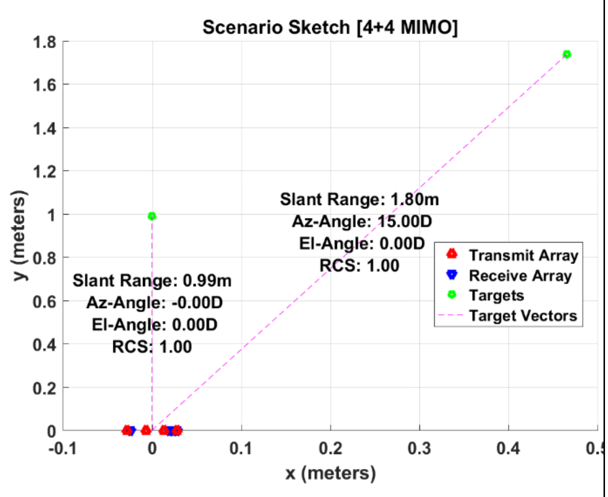


Figure 5-55 Target at 15 degrees; scenario sketch in Cartesian coordinate system

Figure 5-56 Target at 15 degrees; photo of the experimental setup

In Figure 5-57 the range-azimuth map obtained from measurements with the omission of the phase calibration reference and without any post-processing method. A strong reflection at about 15 degrees and 1.85 meters can be observed. This location coincides with the target's actual location.

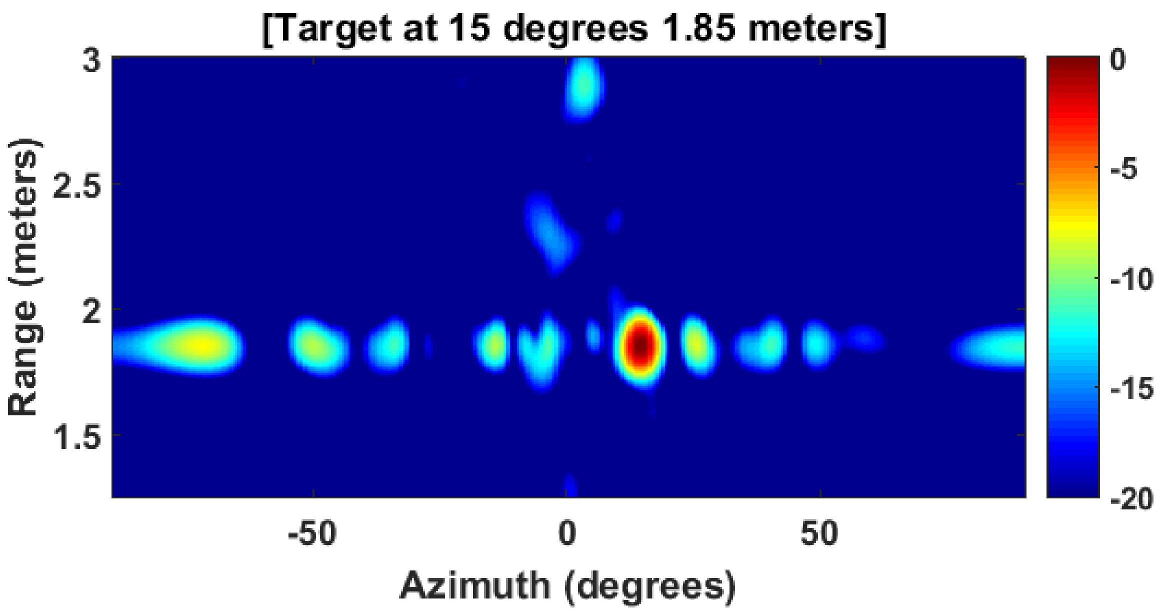


Figure 5-57 Target at 15 degrees; range-angle map (not CLEANed)

In Figure 5-58 the range-azimuth map obtained from measurements with the omission of the phase calibration reference and with CLEAN post-processing. A strong reflection at about 15 degrees and 1.85 meters can be observed. This location coincides with the target's actual location. Apart from that we see that almost all sidelobes and artefacts from the background are gone. However a far target at around 3 meters starts to become visible. It is assumed that this reflection is from the corner of the platform inside the anechoic chamber. But even with the target at the very far which has nothing to do with our test target, this is a good indicator of how well the experimental beam patterns match the expected beam patterns. Because even the target at the far is left alone without its sidelobes.

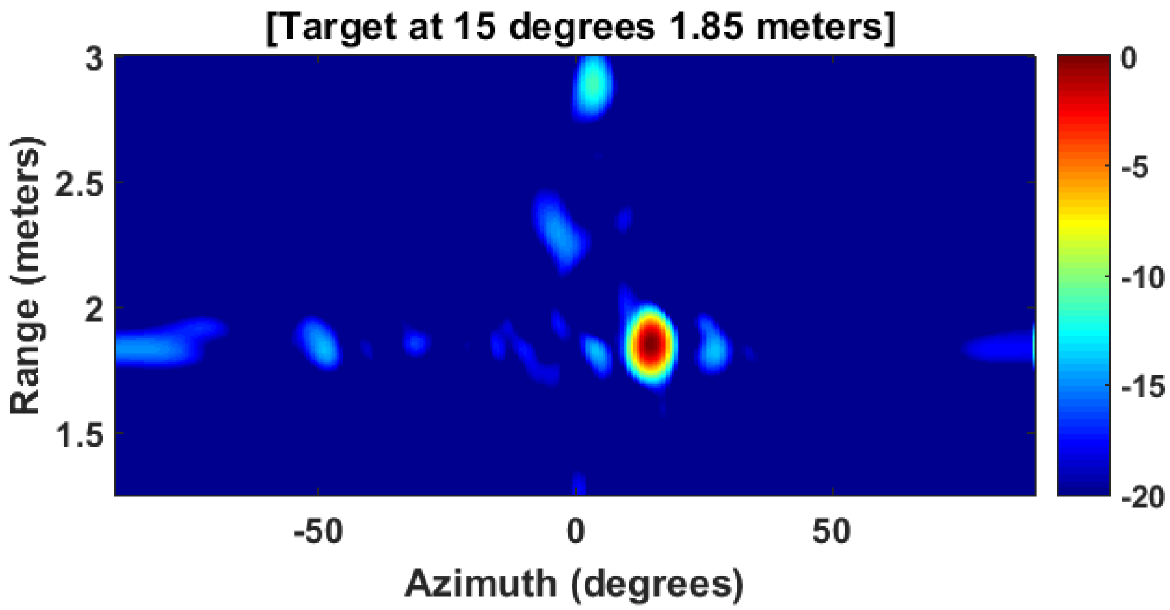


Figure 5-58 Target at 15 degrees; range-angle map (CLEANed)

In Figure 5-59 the azimuth cut from the range-angle map at the target range of 1.85 meters can be seen. In the angle cut beamwidth is measured is about 4.42 degrees, which is well within the 10% vicinity of the simulated beamwidth of 4.61 degrees. Also in the angle cut the sidelobe level can be seen as -7.7 dB which is less than expected compared to simulated level of -10.16 dB, however this can be explained due to noise and other imperfections in the sensor positioning, non-ideality of anechoic chamber and reflections from the reference (i.e. multi-path interference, range and angle sidelobes adding up etc.).

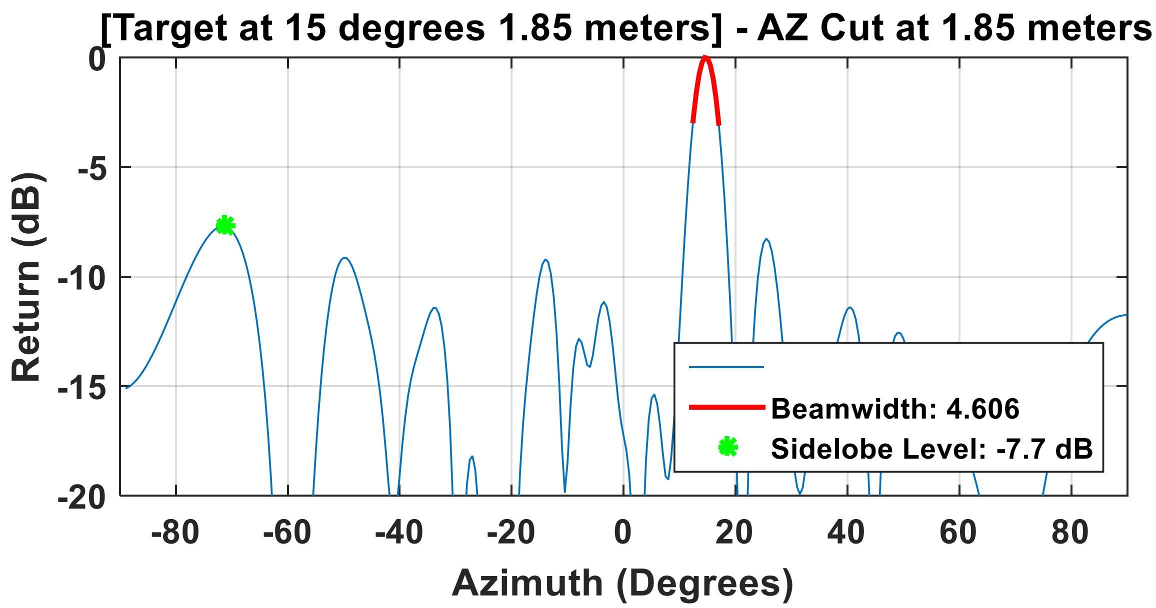


Figure 5-59 Target at 15 degrees; range-angle map (not CLEANed)

<u>*Target at 20 Degrees*</u>

In Figure 5-60 the over the top sketch of the third scenario can be seen. A target is placed at about 1.7m slant range with about 20 degrees angle. The photo of the setup can be seen in Figure 5-61.

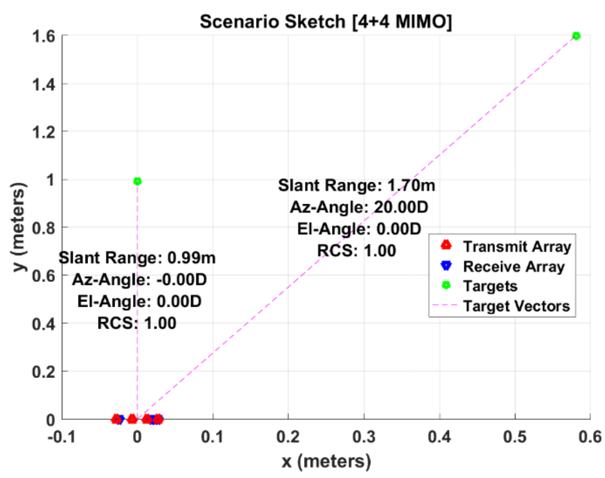


Figure 5-60 Target at 20 degrees; scenario sketch in Cartesian coordinate system

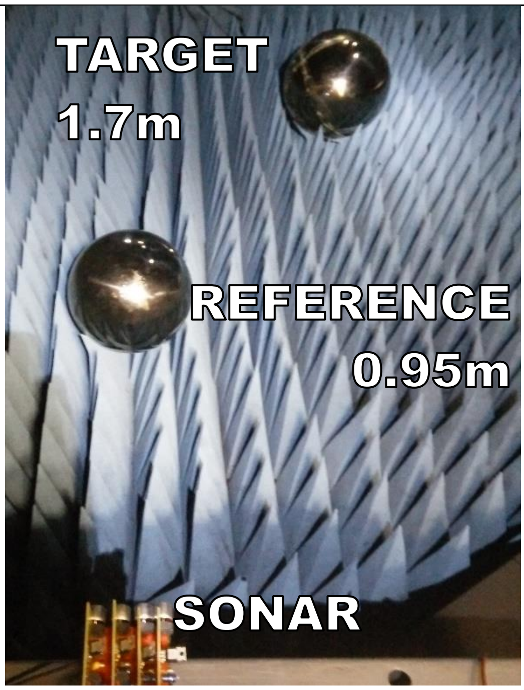


Figure 5-61 Target at 20 degrees; photo of the experimental setup

In Figure 5-62 the range-azimuth map obtained from measurements with the omission of the phase calibration reference and without any post-processing method. A strong reflection at about 20 degrees and 1.7 meters can be observed. This location coincides with the target's actual location.

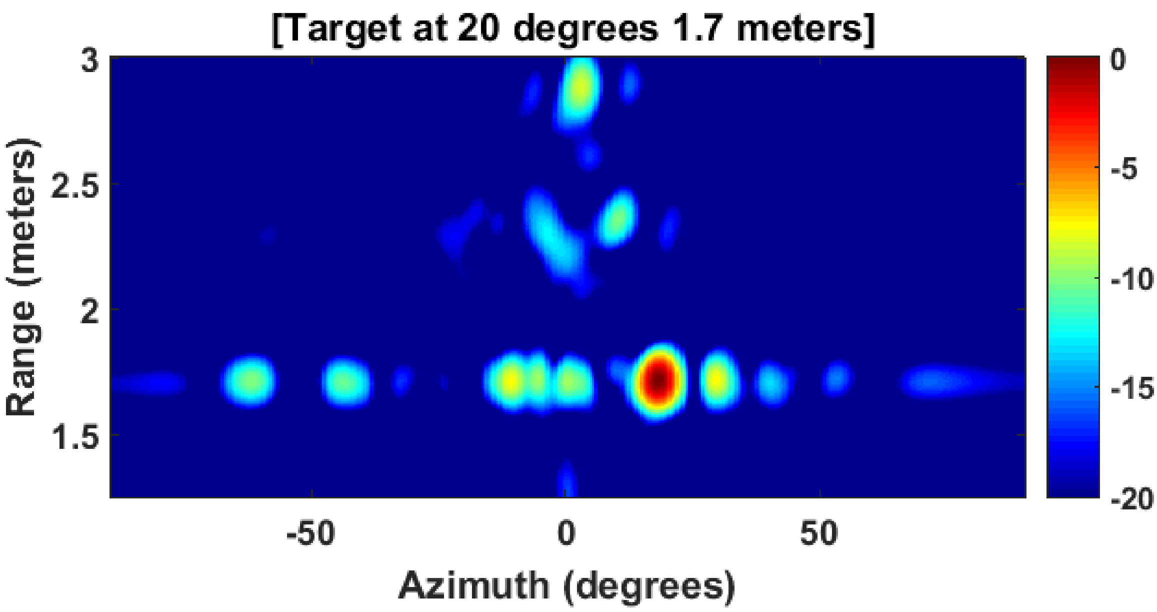


Figure 5-62 Target at 20 degrees; range-angle map (not CLEANed)

In Figure 5-63 the range-azimuth map obtained from measurements with the omission of the phase calibration reference and with CLEAN post-processing. A strong reflection at about 20 degrees and 1.7 meters can be observed. This location coincides with the target's actual location. Apart from that we see that almost all sidelobes and artefacts from the background are gone. The previously mentioned far target at around 3 meters is now stronger. It is assumed that this reflection is from the corner of the platform inside the anechoic chamber, and the reason that it got stronger (relative to our target), is that our target reflection got weaker due to the sensor's beam pattern. There are also 2 other targets that start to show up at around 2.3 meters. It is assumed that these are reflections from the floor of the anechoic chamber due to the slight inclination of the sensors. An increase in the sidelobes can be observed too, this is as mentioned before due to mismatches on experimental beam pattern and the simulated beam patterns. As the incident angle increases, the expected beam patterns starts mismatching the experimental results. This is assumed to be due to approaching the sensor beamwidths and the corruption in experimental beamwidth that comes with it. Even then though, the sidelobe levels can be seen to be around -13 dB, and the target(s) can still be identified without too much space for false alarms.

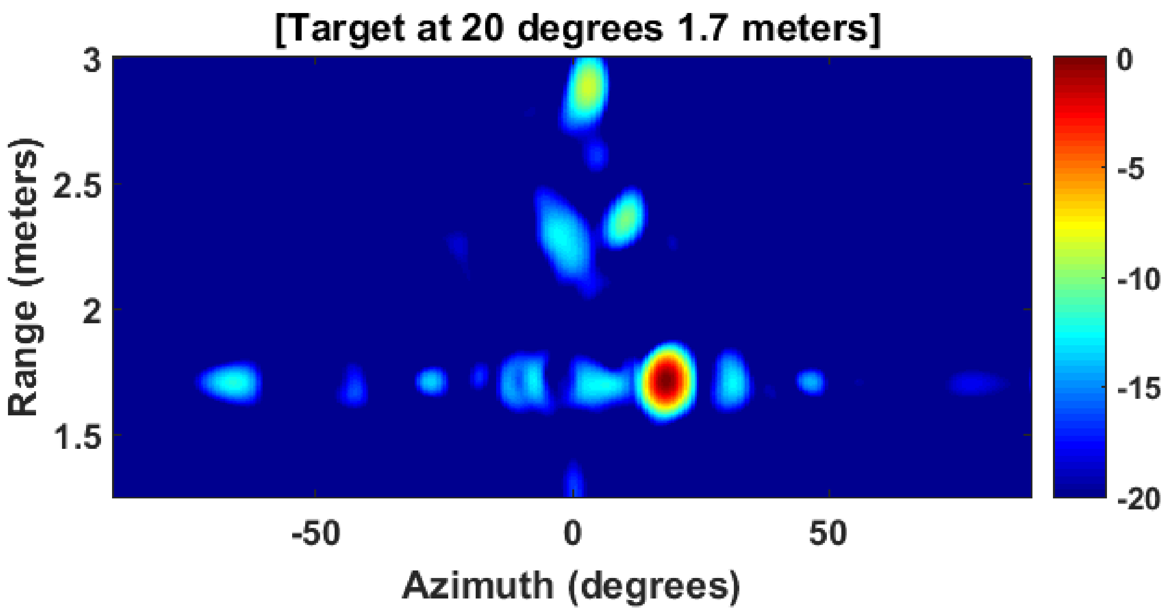


Figure 5-63 Target at 20 degrees; range-angle map (CLEANed)

In Figure 5-64 the azimuth cut from the range-angle map at the target range of 1.85 meters can be seen. In the angle cut beamwidth is measured is about 4.97 degrees, which is still within the 10% vicinity of the simulated beamwidth of 4.54 degrees. Also in the angle cut the sidelobe level can be seen as -7.68 dB which is less than expected compared to simulated level of -10.13 dB, however this can be explained due to noise and other imperfections in the sensor positioning, non-ideality of anechoic chamber and reflections from the reference (i.e. multi-path interference, range and angle sidelobes adding up etc.).

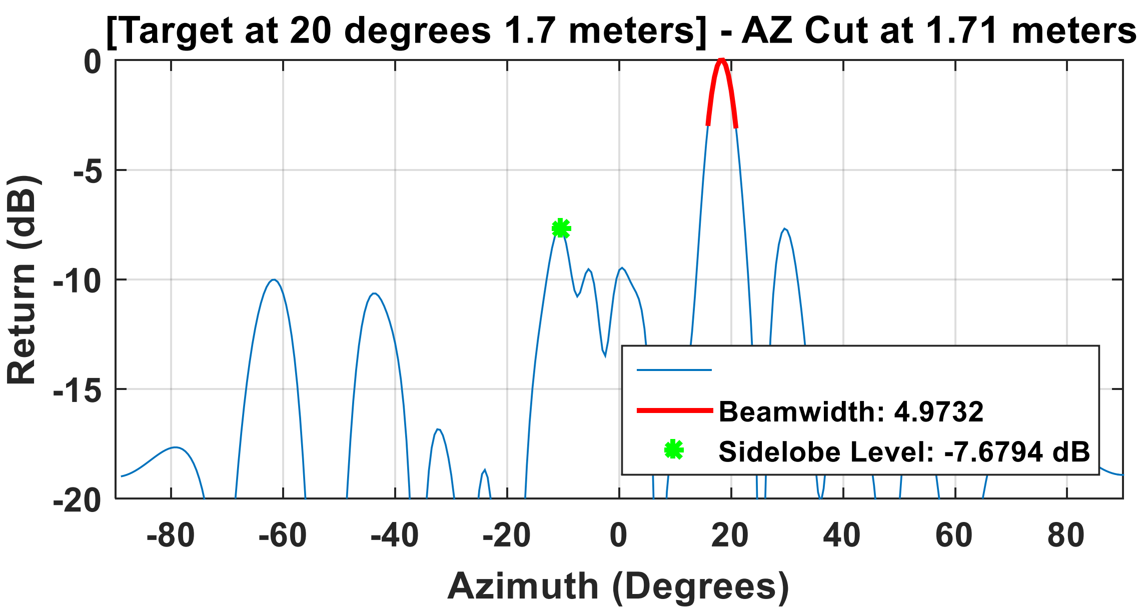


Figure 5-64 Target at 20 degrees; azimuth cut at target range (not CLEANed)

Target at 25 Degrees

In Figure 5-65 the over the top sketch of the third scenario can be seen. A target is placed at about 1.7m slant range with about 25 degrees angle. The photo of the setup can be seen in Figure 5-66.

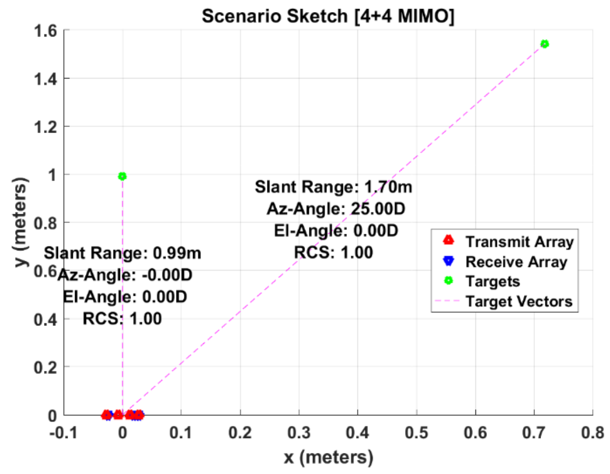


Figure 5-65 Target at 25 degrees; scenario sketch in Cartesian coordinate system

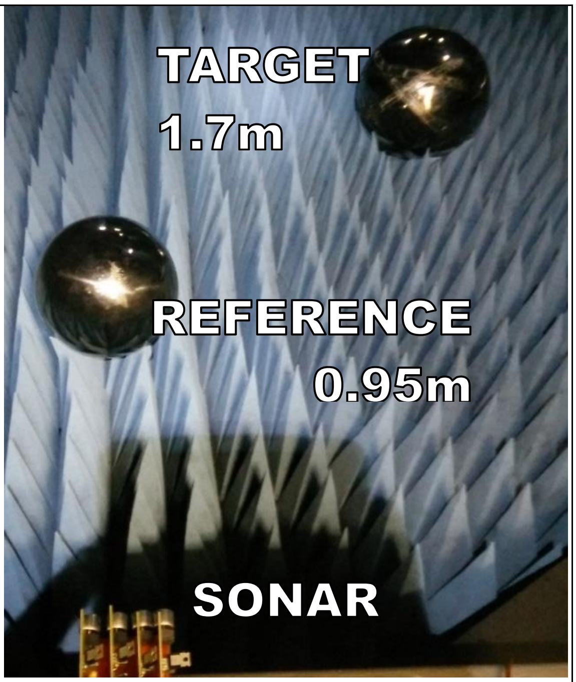


Figure 5-66 Target at 25 degrees; photo of the experimental setup

In Figure 5-67 the range-azimuth map obtained from measurements with the omission of the phase calibration reference and without any post-processing method. A strong reflection at about 25 degrees and 1.7 meters can be observed. This location coincides with the target's actual location.

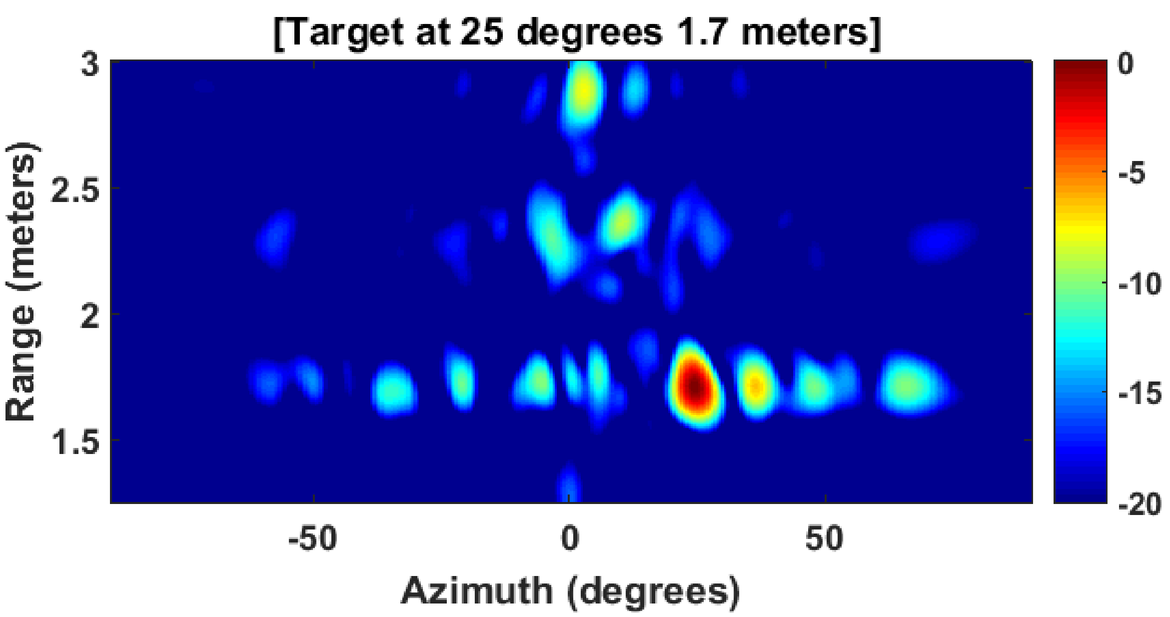


Figure 5-67 Target at 25 degrees; range-angle map (not CLEANed)

In Figure 5-68 the range-azimuth map obtained from measurements with the omission of the phase calibration reference and with CLEAN post-processing. A strong reflection at about 25 degrees and 1.7 meters can be observed. This location coincides with the target's actual location. Apart from that we see that almost all sidelobes and artefacts from the background are gone. The previously mentioned far target at around 3 meters and the other targets around 2.3 are now even stronger. It is assumed that reflection at 3 meters is from the corner of the platform inside the anechoic chamber, the other 2 targets that start to show up at around 2.3 meters are assumed to be from the floor of the anechoic chamber due to the slight inclination of the sensors. The reason that these reflections got stronger (relative to our target), is that our target reflection got weaker due to the sensor's beam pattern. In fact since the sensor beamwidth is only 55 degrees, our target at 25 degrees well reside on the edge or even outside of the operating range of the sensors. Yet, the demonstrator equipment works within expected parameters. An increase in the sidelobes can be observed too, this is as mentioned before due to mismatches on experimental beam pattern and the simulated beam patterns. As the incident angle increases, the expected beam patterns starts mismatching the experimental results. This is assumed to be due to approaching the sensor beamwidths and the corruption in experimental beamwidth that comes with it. Even then though, the sidelobe levels can be seen to be around -13 dB, and the target(s) can still be identified without too much space for false alarms.

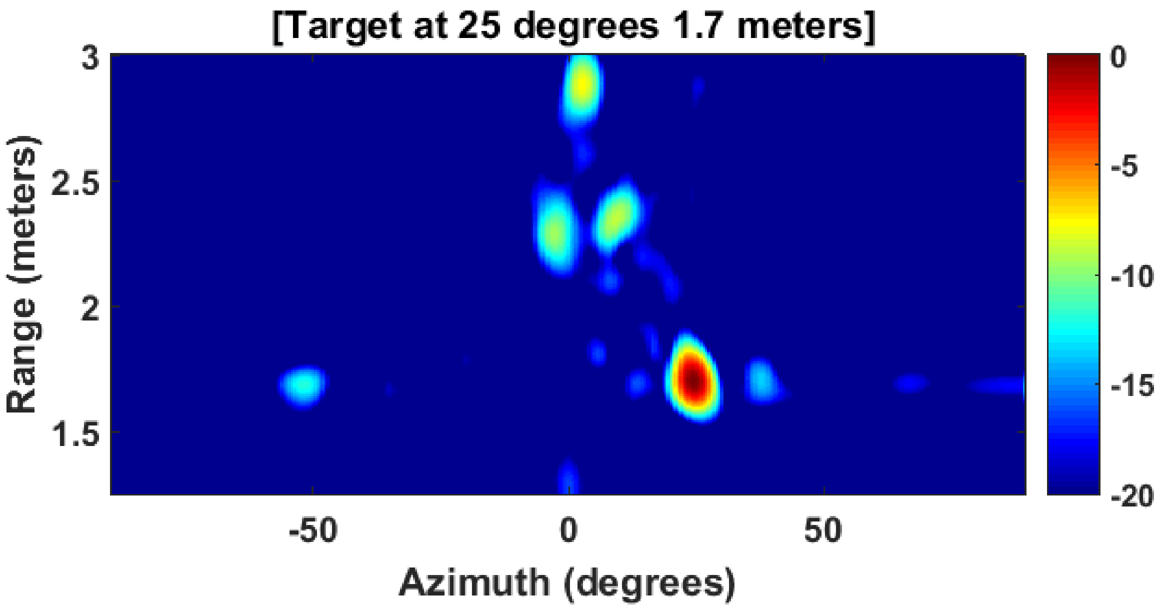


Figure 5-68 Target at 25 degrees; range-angle map (CLEANed)

In Figure 5-69 the azimuth cut from the range-angle map at the target range of 1.71 meters can be seen. In the angle cut beamwidth is measured is about 4.91 degrees, which is still within the 10% vicinity of the simulated beamwidth of 4.72 degrees. Also in the angle cut the sidelobe level can be seen as -6.49 dB which is less than expected compared to simulated level of -10.2 dB, however this can be explained firstly due to the target position which resides outside of the sensor beamwidth and then due to noise and other imperfections in the sensor positioning, non-ideality of anechoic chamber and reflections from the reference (i.e. multi-path interference, range and angle sidelobes adding up etc.).

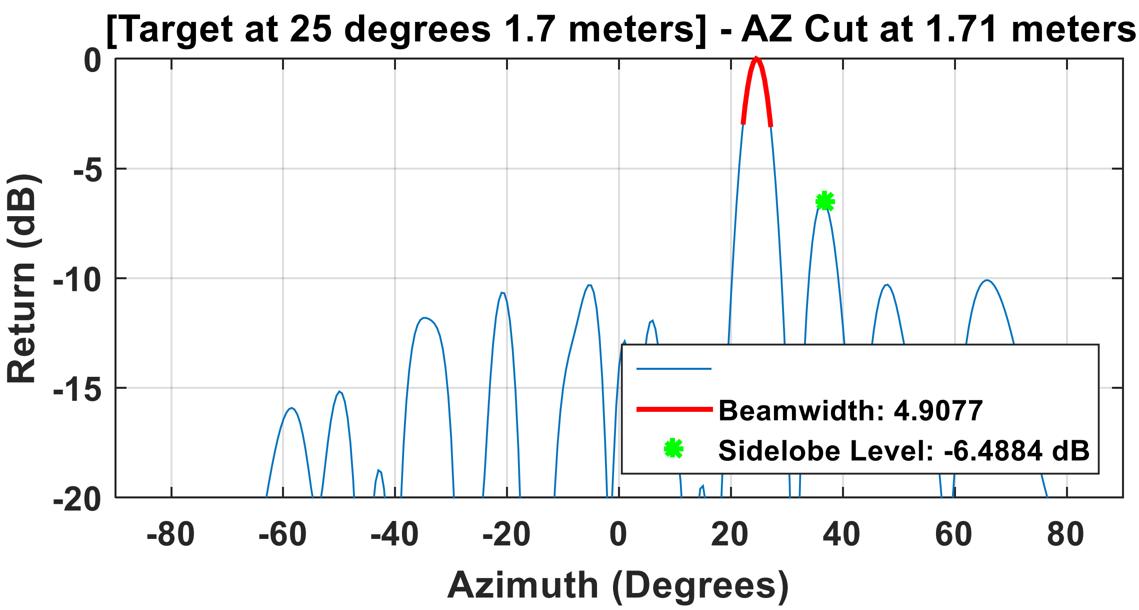


Figure 5-69 Target at 25 degrees; azimuth cut at target range (not CLEANed)

In overall, the experiments yielded favourable results which matched simulations. In summary, the aperiodic MIMO configuration tested has about 4.25 degrees beamwidth only with 8 elements, with around -7.3dB sidelobe levels. This is equivalent to 35% improvement with respect to a conventional 8 element MIMO array or a 16 element phased array. Or in other words it has the equivalent beamwidth of a 24 element phased array.

Experiments with Practical Objects

<u>Target Image of a Bicycle</u>

A mountain bike was placed in anechoic chamber for the purposes of imaging with the technology demonstrator. This experiment was performed to see what a practical and complex shaped object would look like in a range-azimuth map and a range-crossrange map. These measurements were done first with a conventional MIMO array setup like in the previous chapter and the aperiodic MIMO that's developed and tested in this chapter for comparison purposes. The results of these experiments were also super-imposed with an optical image. Below in Figure 5-70 the photo of the experimental setup can be seen. Note that the bicycle has a D-lock mountain in between its tubes, and that there is also a wooden support behind the saddle to give inclination to the bicycle. It is also worth noting that these experiments were done in another anechoic chamber with more space, so reflections from floor, walls or the platforms would be avoided to the best extent.

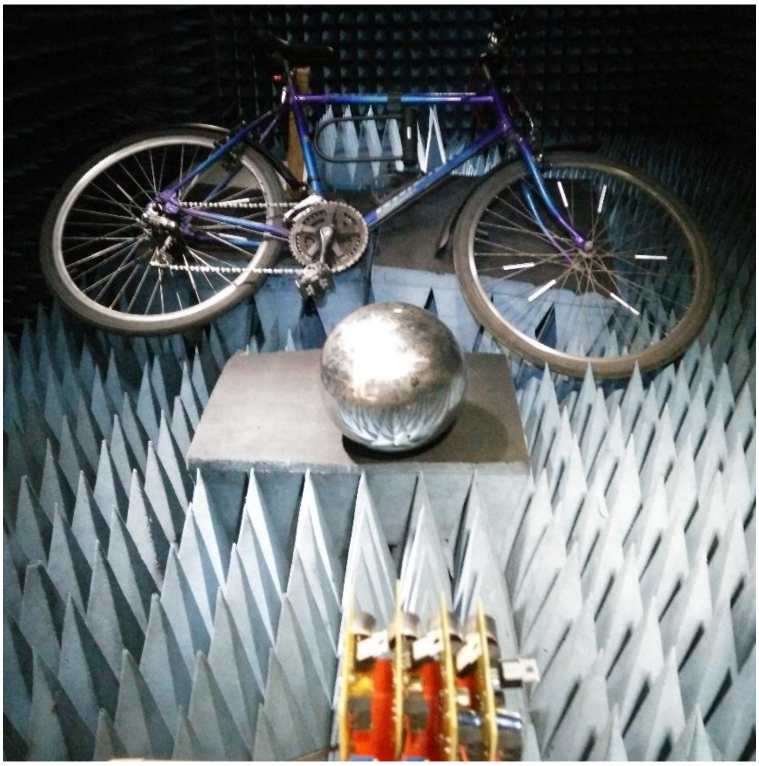


Figure 5-70 Experimenting with aperiodic MIMO with a mountain bicycle; photo of the experimental setup

With current range and angle resolution it wasn't expected to obtain an optical-like image, however a superposition was applied to identify the scattering centres. Below are the range-angle maps obtained from measurements and super-imposed on to the perspective-adjusted optical images. Note that these maps are also passed through CLEAN algorithm to allow better identification of scattering centres.

Arguably, without super-imposing with an optical photo, not much comparison can be made because the acoustic scattering centres of the bicycle are unknown. So these results are skipped in favour of the super-imposed results. But these results can still be found in Appendix .

Below in Figure 5-71 is the first attempt at super-imposing a stretched-to-perspective optical image with the range-crossrange map obtained via experiments with conventional MIMO sonar as designed and used in previous chapter. The existence of a reference target was used to advantage during this process and helped align the sonar image to the optical image. However due to the strong reflection from the reference target, the observation of bicycle's profile became more difficult since its reflectivity was mostly supressed by the reference target. For this reason another range-crossrange map omitting the reference target was produced and super-imposed to the optical image using the same dimensions.

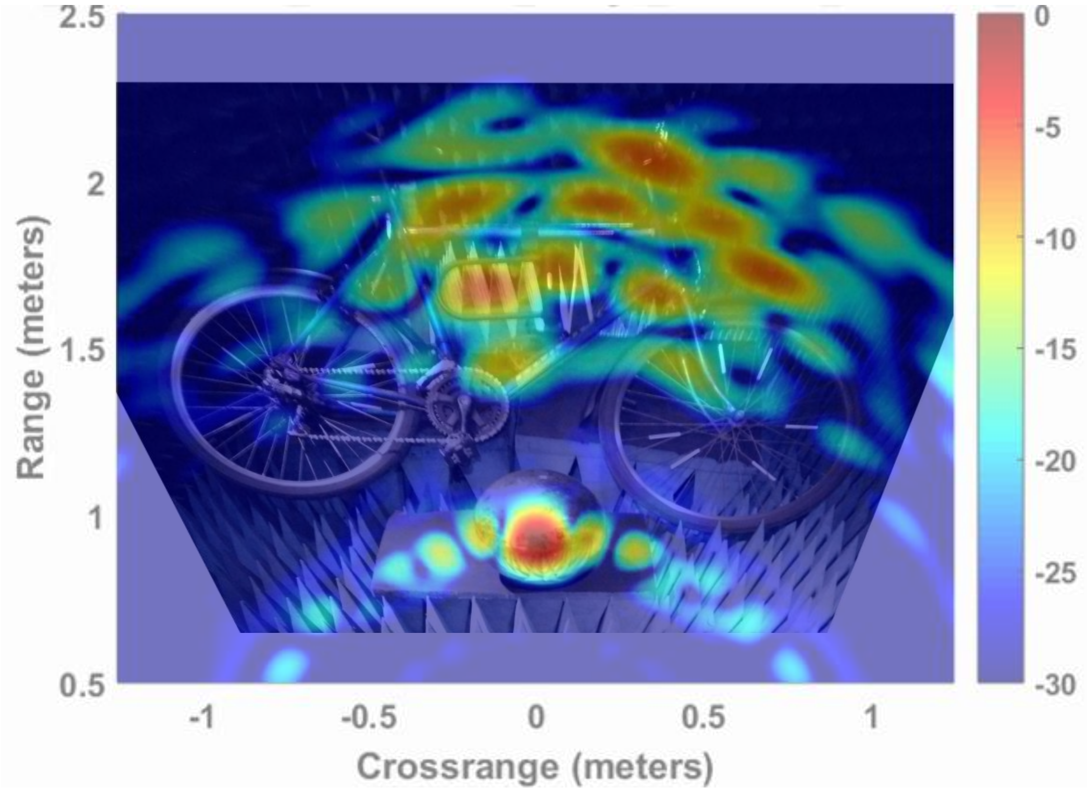


Figure 5-71 Super-imposed raw range-crossrange map of mountain bike with reference target obtained via experiments with conventional MIMO sensor array

Below in

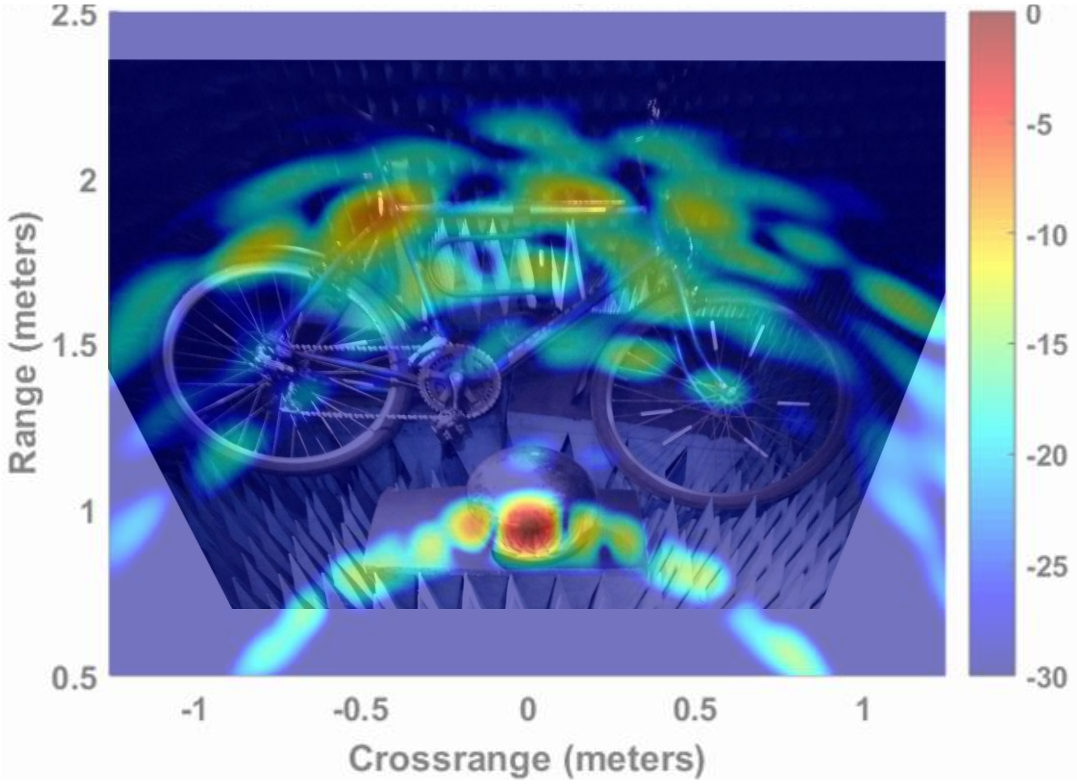


Figure 5-72 is the attempt at super-imposing a stretched-to-perspective optical image with the range-crossrange map obtained via experiments with aperiodic MIMO sonar as designed and used in this chapter. Again, the existence of a reference target was used to advantage during this process and helped align the sonar image to the optical image. But this processed also suffered from the strong reflection from the reference target; the observation of bicycle's profile became more difficult since its reflectivity was mostly supressed by the reference target. For this reason another range-crossrange map omitting the reference target was produced and super-imposed to the optical image using the same dimensions.

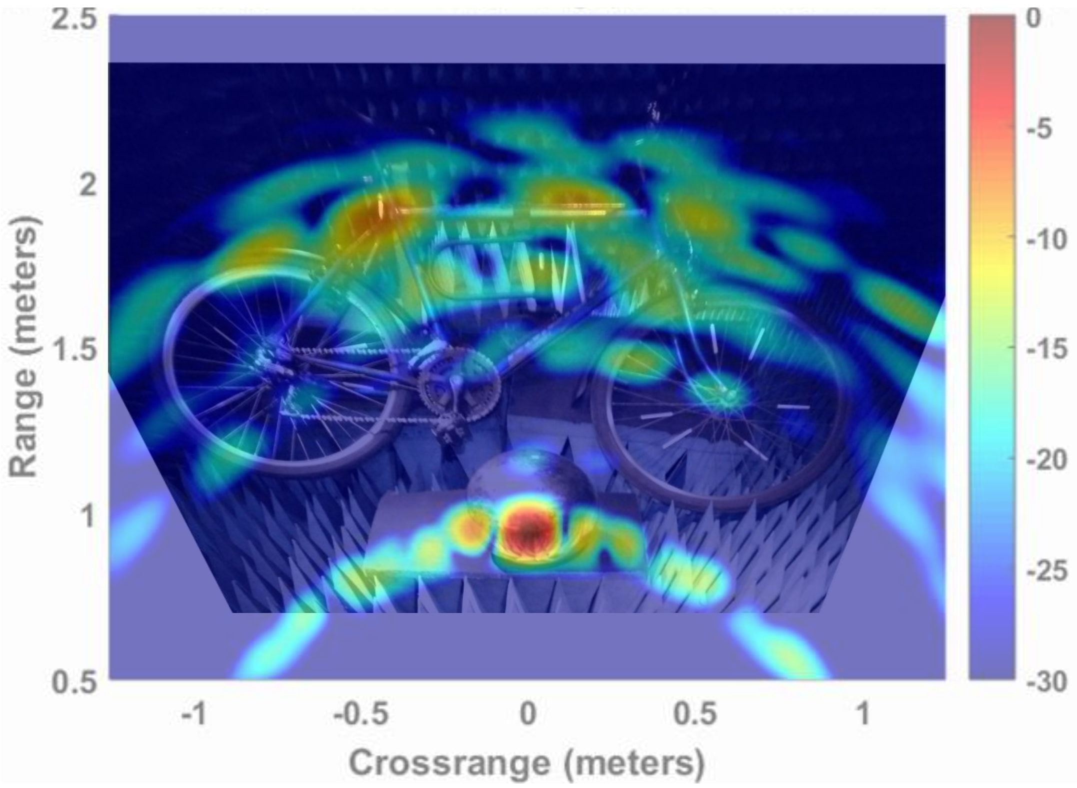
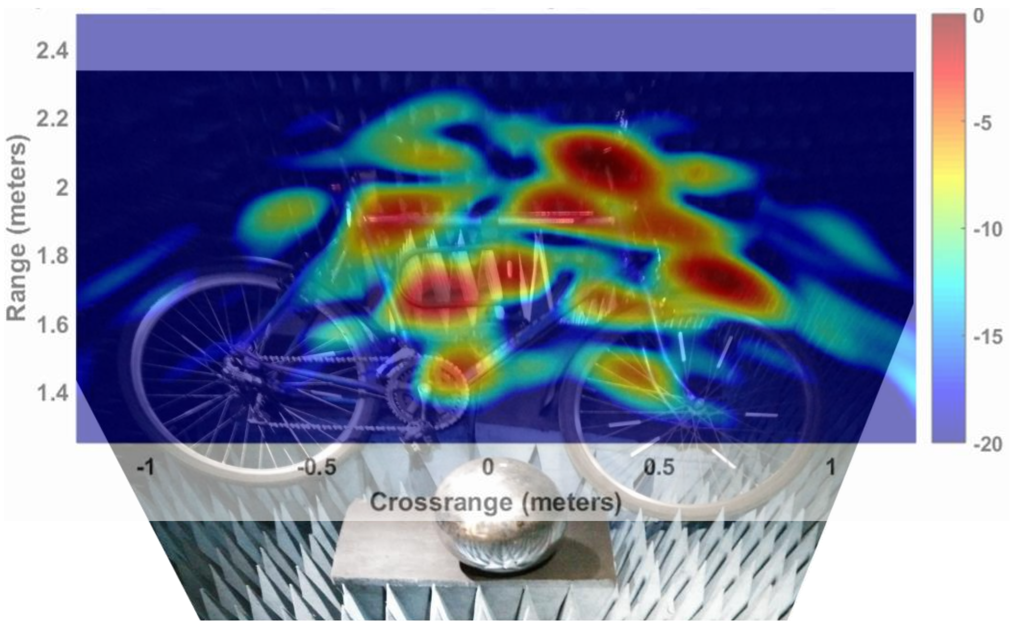


Figure 5-72 Super-imposed raw range-crossrange map of mountain bike with reference target obtained via experiments with aperiodic MIMO sensor array



Below in Figure 5-73 the super-imposed sonar image taken with a conventional MIMO sonar omitting the reference target can be seen. When the measurement with a conventional MIMO is observed, high reflections can be seen from the handlebars, horizontal frame tube, bottom of the saddle, and thick part of the D-lock hanging from the horizontal tube, thin part of the D-lock, gear rings, and left side of the front tyre.

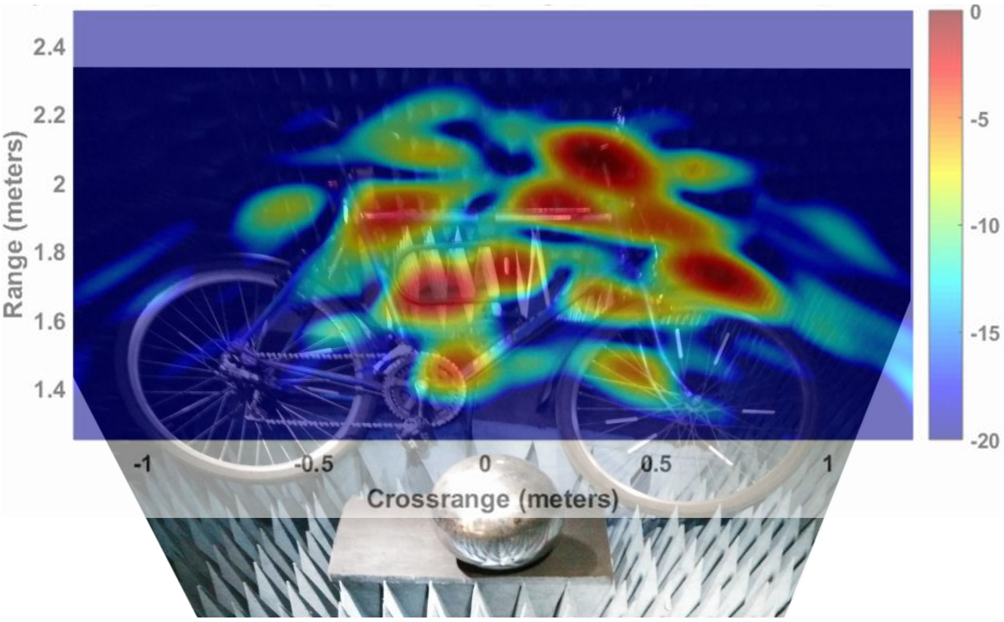
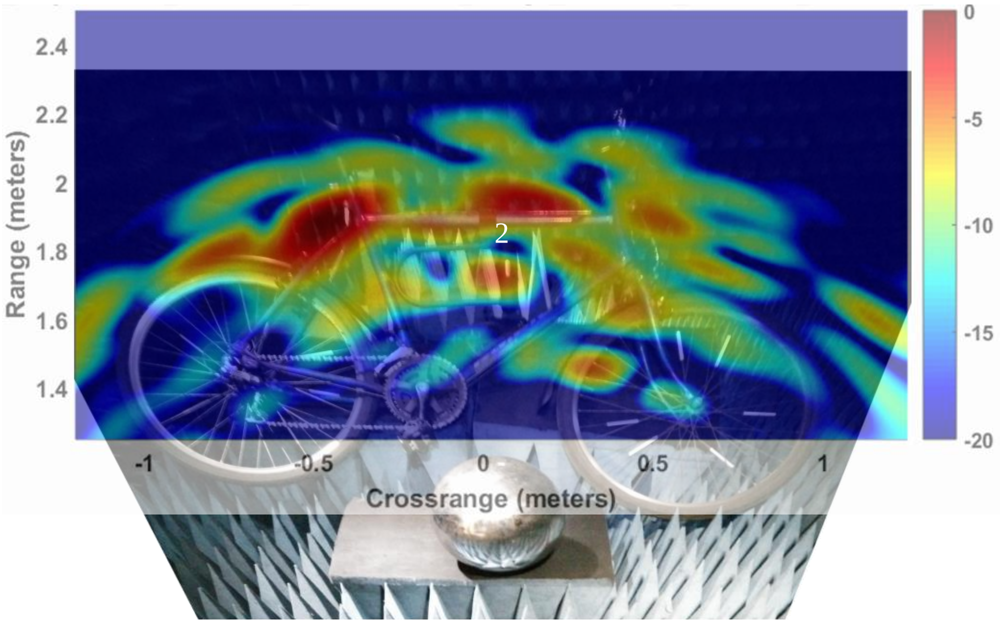


Figure 5-73 Super-imposed CLEANed Range-crossrange map of mountain bike obtained via experiments with conventional MIMO sensor array



Below in
Figure 5-74 the super-imposed sonar image taken omitting the reference target with aperiodic MIMO sonar can be observed. In comparison, although the range-angle maps look very similar, there are some key differences.

When the handlebar area is inspected, more details and more centres can be seen; such as both the left and the right of the handlebar. The area mentioned is specifically 0.5 meters crossrange and 1.9 to 2.1 meters.

If the D-lock is to be observed, the main reflection from the thick part can still be found in the same location. But now, the reflection from the thin part is actually separated from three other centres; the front of the rear mudguard, the back vertical tube, and the thin part of the D-lock. The area mentioned is specifically 1.7 meters range and -0.4 meters to 0.1 meters crossrange.

When the reflections from the gear rings are inspected, we see that it resolves into three separate centres; the rings, the bottle cage and the front diagonal frame tube. The area mentioned is specifically 1.5 meters range and -0.25 meters to 0.2 meters crossrange.

Similarly if the reflection from the front tyre is inspected, it can now be seen that it also resolved into two separate centres. One originates from a more reasonable source (compared to the thin tubes of the front tyre); the rear mudguard and the back of the front tyre. Another one originates from the centre of the front tyre. Tyre mentioned is specifically around 1.5 meters range and 0.3 meters to 0.6 meters crossrange.

Also, when the rear mudguard is inspected, it can now be observed that a reflection which wasn’t quite visible before. The area mentioned is specifically 1.8 meters range and -0.8 meters crossrange.

Finally when the middle of the horizontal tube frame is to be inspected, a new reflection centre can be observed, which is resolved from the reflection from saddle and the right side of the horizontal tube. The area mentioned specifically is 1.9 meters range and -0.3 meters to 0.4 meters range. Also because of this resolution, the reflection from the saddle can be seen shifting to left and actually originating from the left merging point of the vertical, horizontal and diagonal frame tubes. The area mentioned specifically is 2.1 meters range and -0.4 crossrange.

In comparison the target strengths may seem to be decreasing, but this is only because of same reflection power being redistributed over more centres as mentioned before.

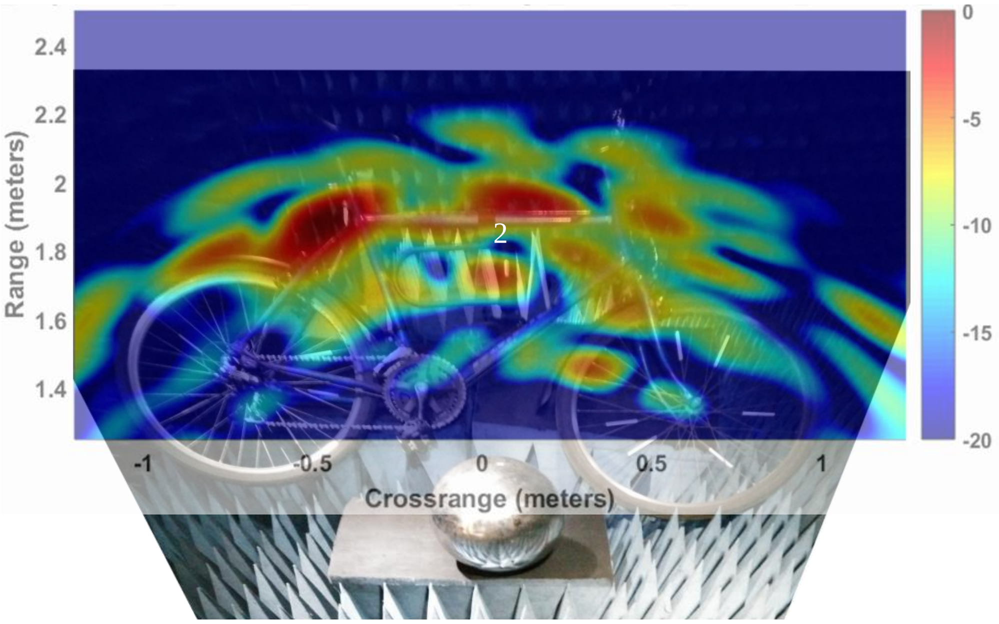


Figure 5-74 Super-imposed CLEANed Range-crossrange map of mountain bike obtained via experiments with aperiodic MIMO sensor array

5.6 CONCLUSIONS

Further performance optimisations via aperiodic MIMO configurations were explored via use of heuristic optimisation algorithms. Three of the well-known and commonly used algorithms were modified to address our constraints and performance variables. A configuration obtained via heuristics was tested via simulations and a housing for the sensor array was 3D printed. This configuration then was tested in anechoic chamber and its performance was confirmed in practice. The aperiodic MIMO configuration that we have tested has about 4.25 degrees beamwidth only with 8 elements, with around -7.3dB sidelobe levels. This is equivalent to 35% improvement with respect to a conventional 8 element MIMO array or a 16 element phased array, and it has the equivalent beamwidth of a 24 element phased array. After experimental verification with calibrated targets, both conventional and aperiodic MIMO technology demonstrator was also put into test to measurement of a practical object (a mountain bicycle) for comparisons. Results were compared highlighting the improvements. With the aperiodic MIMO, more scattering centres were observable. In comparison, improved angular resolution helped separate merging scattering centres, therefore yielding more detail about the target.

Even though it was not initially planned to look at Doppler applications with MIMO arrays, next chapter will look into MIMO MTI (Moving target indicator) applications. The Doppler information of moving targets may be used to better classify targets, or the Doppler characteristics of a moving platform can be used to compensate for better beamforming during vehicular motion. So this study would be a pre-cursor to more specific Doppler applications with the use of MIMO sensor arrays.

6 MIMO MTI APPLICATIONS

The work under this section was done under another project called SIMITAR (Persistent Surveillance from air with a low frequency MIMO Towed Array Radar) which was funded by DSTL. Exploring MTI applications was not initially under this research's scope so applicability of MTI was not considering during the choosing of ultrasonic or RF for the technology demonstrator in the previous chapters. And so MTI applications could not be explored with the ultrasonics and the technology demonstrator due to inability to use ultrasonic on a vehicle at speed. Also, due to the requirements of SIMITAR, ultrasonics was not an option for MIMO medium at all. However, SIMITAR's research and findings of MIMO MTI applications were still thought to be extremely useful and relevant for automotive applications and were thought it could be a precursor for exploring automotive MIMO MTI applications and therefore included in this thesis.

6.1 INTRODUCTION

When a wave of any medium hits a moving target, the scattered wave shows Doppler characteristics. the scattered wave's frequency will shift depending on the target's velocity vector with respect to the receiver (i.e. norm of the velocity vector with respect to the receiver centered position vector of the target). The Doppler Effect can and widely used for measuring target radial speeds. In some special cases where targets might be close to or even under strong non-moving surrounding, the Doppler Effect can be used to filter out and detect targets even if their target signature might not be strong enough to surpass clutter levels. This application could be especially useful in an automotive scenario to detect and even classify other motorized vehicles, bicycles and pedestrians.

Moving target indication application (MTI) can be done with –among other systems- a multi-pulse sensor system. Since the remainder of this research focused on pulse transmitting MIMO arrays, the focus of this chapter will focus on multi-pulse MIMO sensor array Doppler applications. The previously used experimental systems were already using multiple pulses to improve SNR, but the collected data was not processed in any other meaningful way to extract more information. That is; the pulses were simply used for a coherent sum to increase SNR.

6.2 Moving Target Indication (MTI)

In a multi-pulse system, samples from same range/time points should return the same phase if the given range/time points do not move [9]. In such case if a Fourier transform was to be taken alongside the slow time domain (i.e. slow time domain was to be converted to frequency domain), the –Doppler- frequency of the entire range/time samples would be zero due to non-changing phase. However, if a target is moving inside the same range resolution bin, then this target's return phase would change with every pulse (i.e. in the slow time domain). This is basically what causes the Doppler Frequency shift. And in such case if a Fourier transform was to be taken alongside the slow time domain, then moving targets would show a frequency other than zero, which would then allow us to separate them from other strong returns. Note that, how much a target's phase would change is not only dependent on its radial speed but also the time between pulses (i.e. PRI or PRF).

In cases of strong foliage, the return from zero-Doppler band (stationary targets), could still be too strong that the moving targets -which are likely to be much weaker- may still not be visible due to dynamic range limitations. In MTI applications, as the name suggests, the focus is usually on moving targets; therefore it is also common application to apply a high-pass filter to received signals in either slow-time domain or in Doppler domain. The appropriate application of a high-pass filter, would ideally eliminate all returns from stationary targets, foliage and background leaving only the moving targets to be identified.

Most of the aforementioned practice however is only available assuming the system has a narrow beam. A relatively narrow beam is often a requirement for most Doppler applications. The fact that this research's focus was on developing narrow beam sensing technologies, combined with our practices that we're already employing coherent multi-pulse sensing, already fulfils the requirements for using MIMO sensor arrays for Doppler applications.

6.3 PLATFORM MOTION COMPENSATION

In some applications such as airborne or automotive, the assumption that sensor platform is stationary may not always be correct. In such cases, the radial speed of the possible targets would be measured with respect to the platform motion. However, often times what an operator needs is the radial speed information with respect to a ground-stationary body. Apart from this phenomenon, the ground reflections which are supposed to be at zero-Doppler band also shifts creating a ground-clutter band outside the usual region. Due these reasons and similar, a need for compensating the speed of the sensor platform arises.

In single narrow beam applications the compensation process is rather straightforward. The platform's Doppler frequency with respect to where beam is pointed can be calculated if the platform velocity with respect to that point is known. Using that Doppler frequency, received signals then can be either phase shifted in time domain or simply shifted(subtracted) in Doppler (frequency) domain.

In multi-beam applications, this process gets a bit trickier since the beam direction is not fixed. This means each beam will end up having a different Doppler return from any ground-stationary targets. One of the simplest methods to approach this problem is; to beamform first, and treat each output beam as if they're separately measured single-narrow-beams and apply the aforementioned platform motion compensation logic. Code segment which accomplishes this task can be found in the Appendix .

6.4 SIMULATIONS AND EXPERIMENTS

The experimental setup for this chapter was substantially different than previous what was used in previous chapters. To look into MTI applications with MIMO radars was only decided later on during research, hence possible MTI applications were not considered during the previous design phases. In order to be able to test MTI applications with MIMO sensor arrays, new experimental setups were implemented. There were two different experimental setups that were used. First setup was one involving a VNA with a TDMA setup which required moving of the sensors for each transmit/receive pair (similar to what's been implemented in previous chapters). And second one was a setup involving a full 4-by-4 MIMO array radar operating at 435 MHz carrier frequency, also using a TDMA scheme.

First setup -also referred to as the indoor setup- was using a 15GHz centre frequency (20mm wavelength) with 1GHz bandwidth (15cm range resolution). Measurements were made with a VNA and results were collected in frequency domain. These results were then transformed into time domain and used as if they were returns from a chirp signal with 15GHz centre frequency and 1GHz bandwidth. The indoor setup allowed us to mimic movements of targets and/or platform in a precise manner that allowed us to make measurements with expected parameters (such as the platform speed, target positions and speeds).

Second setup –also referred to as the outdoor or rooftop setup- was using a 435 MHz centre frequency (70cm wavelength) with 4 MHz bandwidth chirp signals (37.5m range resolution). Measurements were made with a pulse coherent MIMO array radar and results were collected in time domain. This setup involved a non-controlled environment as suggested by its nature, and used targets of opportunities. The verification of measurement were made by comparisons to optical videos of the measurement period.

6.4.1 Simulations

A simulator was programmed in MATLAB to exactly mimic the hardware and software blocks of both the systems. First reason for the development of this tool was to have a basis for developing the signal processing algorithms for the system with an expected and controlled environment. Second reason was to observe the radar signals and to learn about what to expect during transmit and receive. The simulator programme was in fact a modified version of the simulators used in previous chapters with one major modification. The signal simulation was done on baseband level compared to carrier level in previous chapters, this gave us flexibility in simulating our different setups used in this chapter.

Simulator program uses the exact same waveforms transmitted from each transmitter during experiments and modifies their amplitude, time delay, phase, signal-to-noise ratio and other properties in software to mimic radar signals collected via each receiver. The end product of the simulator program is the exact same type of output obtained from our experiments. This separation of simulation and processing allowed for development of signal processing algorithms, testing and verifications of them through simulations and emulations and then use them for experiments to verify overall system functions.

In order of execution, below are bullet points summarising the simulation's key aspects:

- MIMO array definition (number of elements, element positions, platform altitude)
- Target definitions (3 dimensional positions and velocities)
- For each frame;
    - For each transmitter;
        - And for each receiver
            - Simulate the transmitter to receiver direct path signal
            - For each target
                - Time delay the signal with respect to exact range from transmitter to target and then to receiver
                - Apply phase shifts to signal with respect to range
                - Attenuate signal strength with respect to range (free space propagation)
            - Sum the separately modified direct path and target reflection signals as if they were added in space
            - Mimic the analogue amplifier and filtering circuitry

The simulator in the end returns streams of data as if they're captured from the receiver antennas. This data is then fed into the signal processor part of the program. Signal processing system deals with range gating, matched filtering, beamforming, Doppler processing and data visualisation.

6.4.2 Signal Processing

Firstly, all the data streams from receivers are passed through matched filters corresponding to transmit waveforms. Note that at this testing stage all transmit waveforms are the same which is a linearly modulated frequency signal. The orthogonality is provided by TDMA. After matched filtering the first high peak is assumed to be the direct path signal and any data before that is discarded.

Second step is range gating; a safe portion from data is discarded (i.e. time of flight for waves to reach from platform altitude to ground, and ranges outside the power budget) to avoid unnecessary computation load and minimise the effects of direct path transmission.

Third step in processing is the initial Doppler processing. The virtual array element signals would be transformed into Doppler frequency domain from slow-time to obtain a preliminary look on range-Doppler data without any angle information. Only reason that this is done to ease the data presentation process as will be explained in the last step.

Fourth step in processing is the beamforming. Individual frames of all virtual array elements are used to create range-angle vs reflectivity data. Beamforming is done using a sum-beamformer while considering a known platform altitude. The end result is multiple frames of range-angle maps.

Fifth step in processing is the Doppler processing. The separation of range-angle maps is in slow time, therefore a Fourier transform among the range-angle maps yields range-angle maps in Doppler domain. Since it makes it easier to understand and visualise the data, the dimensions of the data is permuted to have range-Doppler maps in different scanning angles. Range-Doppler maps are then subjected to the platform motion compensation logic introduced before. And finally, the data is then filtered to cut out zero Doppler band to reject direct path transmission, ground clutter and other ground-stationary targets.

Sixth and final step is the data visualisation. Last block yields us range-Doppler maps in different scanning angles. One method used to visualise this data is to plot heat maps of range-Doppler maps in scan angles of interest (in order to avoid flooding with plots). Another method used is to again plot heat maps of angle-Doppler maps in range resolution bins of interest, which is good for resolving and counting multiple targets in the range resolution bin.

Final and most simple method of presenting range and Doppler information without going into angle information was to draw heat maps of range-Doppler data without any beamforming on a virtual array element basis. So a single range-Doppler map would be obtained from each virtual array element and magnitudes of these maps would be summed to increase SNR. Only way that this could be done was to have another Doppler processing block before beamforming. This was found to be the most intuitive way of presenting results, to summarise the flow of information would be as below:

1. Presentation of a single range-Doppler map (MIMO MTI maps as we named them)
2. Identification of moving targets and choosing range-Doppler points of interest
3. Looking at the magnitude-angle plots at selected range-Doppler points of interest.

6.4.3 Experimental Setup in 15GHz Band

For indoor experimental measurements a 2-port VNA was used a radar transceiver. VNA was set to sweep a number of frequencies and the complex frequency response was measured within a selected band. The collected complex frequency response then transformed into a time-domain signal using inverse Fourier Transform and was treated like it was a measurement made by transmitting a chirp signal at critical sampling (i.e. complex sample rate equal to bandwidth).

VNA was used to emulate each transmitter and receiver pair in a TDMA scheme. So in comparison to previous chapters, the setup was similar but in this setup, transmitter was also moved in between pulse transmissions in addition to receiver. So number of pulses sent for a single measurement was the multiplication of number of transmitters and number of receivers. A single measurement taken like this is referred to as a frame from this point forward for the sake of simplicity.

Two wide-band and wide-beam horn antennas were connected to VNA ports and moved precisely with digital linear positioners between pulses to emulate a MIMO array. The target was also on a linear positioner of its own and was programmatically moved in between frames to emulate a constant and controllable target radial speeds. The use of linear positioner for transmit and receive antennas also allowed to move the platform in a specific manner in between frames, therefore allowing us to emulate platform motion as well. A photo of the measurement setup can be seen in Figure 6-1.

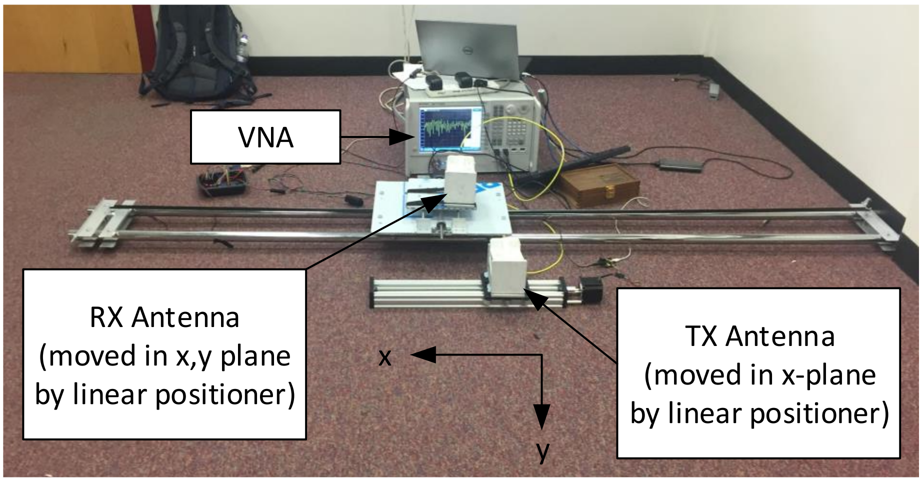


Figure 6-1 Photograph of radar setup for experiments at 15GHz.

For the clarification of moving directions and a better understanding of the setup a sketch has been drawn and can be seen in Figure 6-2. The transmitter and receiver were put on different rails both of them capable of moving the transmitters in the direction of the array line-up (designated X direction). Targets were placed in the Y direction with relatively small angles in order to prove left-right resolving capabilities. Target motion was also emulated strictly on Y direction, which meant due to target positioning it wasn’t necessarily on a perfect radial. Apart from a moving target, a stationary target was also placed in the setup for referencing and verification.

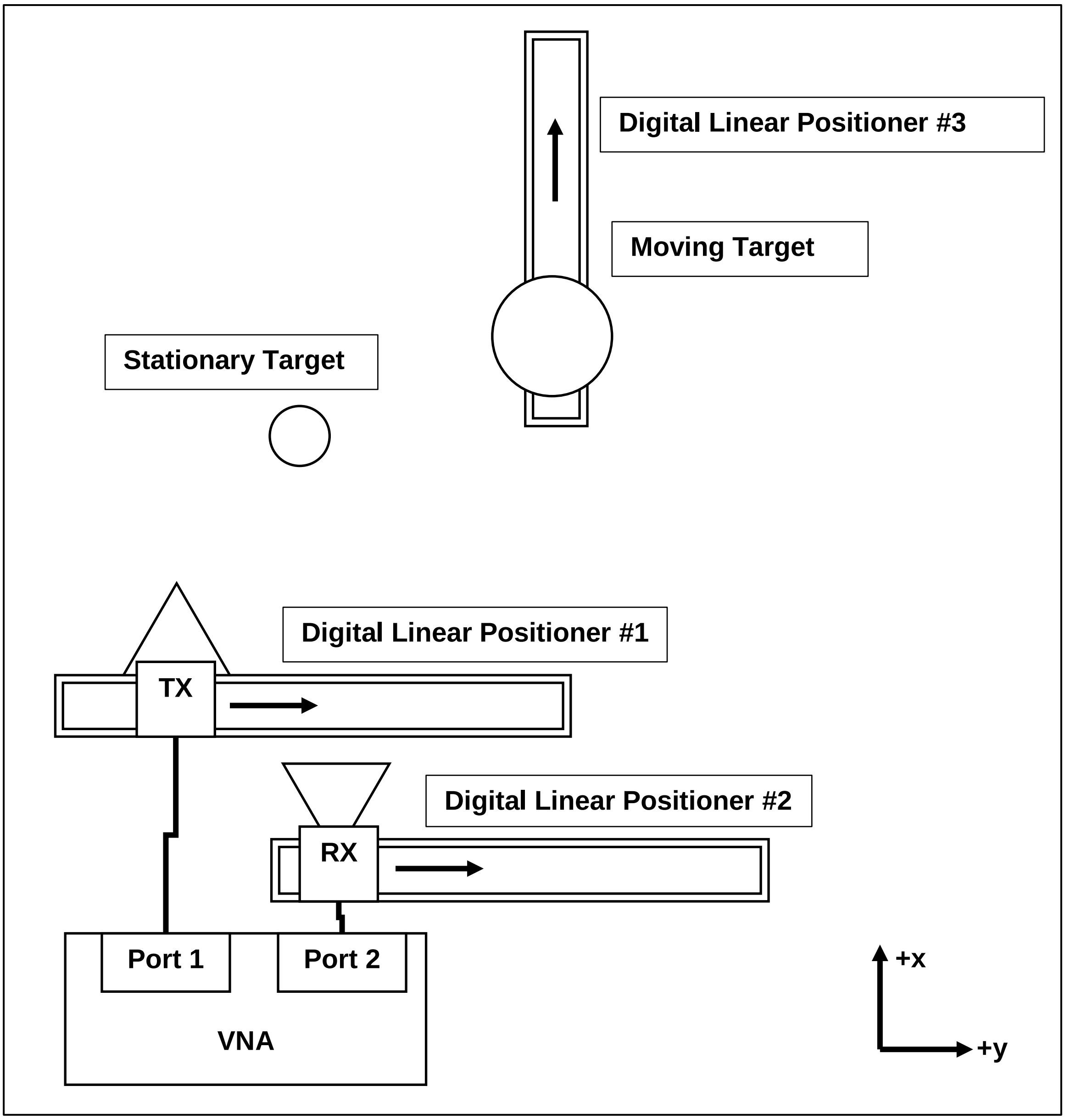


Figure 6-2 Setup for SIMITAR experiment at 15GHz (not to scale)

Below in Table 6-1 the experiment parameters can be found. Most of the parameters were kept same or similar compared to previous experiments.

Table 6-1 VNA settings for VNA used for the experiments at 15GHz band

Experiment Property	Value	Units
Number of Tx	4	-
Number of Rx	4	-
Carrier Medium	RF	-
Carrier Frequency	15	GHz

Wavelength	20	mm
Waveform	Emulated chirp	-
Bandwidth	1	GHz
Range Window	Gaussian (alpha=1.5)	-
Angular Window	None	-
Tx Sensor Beamwidth	120	Degrees
Rx Sensor Beamwidth	120	Degrees
Number of Points	201	-
Tx Element Spacing	40	mm
Rx Element Spacing	10	mm
Fraunhofer Range	>2.5	m
Complex Sample Rate	1	Gigasamples/s
Tx Power	10	dBm
IF Bandwidth	1	kHz
PRI (assumed)	4	ms

The array elements were positioned in a conventional MIMO array setup with a coarse transmit array. It was decided to emulate 4 transmitters and 4 receivers since this was the setup that was used in previous chapters too. The receivers were positioned with half-wavelength spacing whereas transmitters were positioned with double-wavelength spacing to match the fine receiver positioning. Due to physical limitations some space in Y direction were left between the transmitter row and receiver row. The emulated array configuration and its resulting virtual array configuration can be seen in Figure 6-3 sketched in terms wavelength.

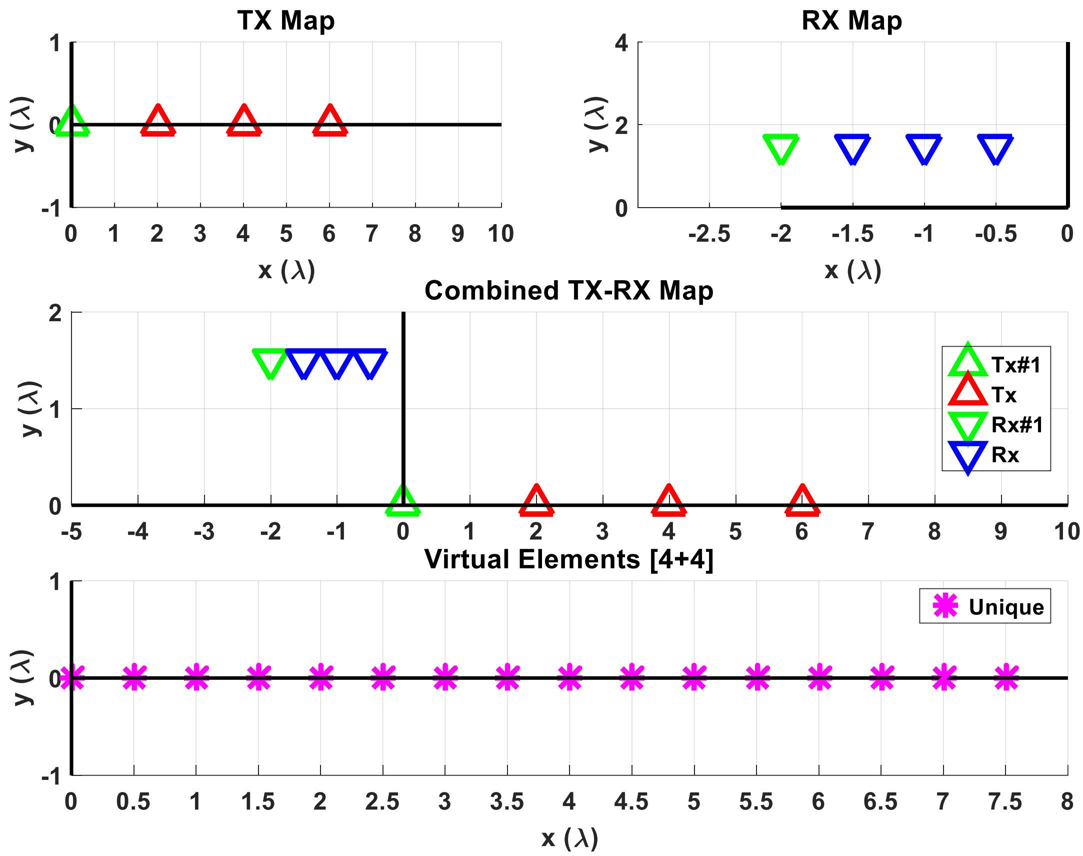


Figure 6-3 Antenna positions of MIMO array in term of wavelength and its virtual array position.

Since the motion of targets and platform were emulated, the real time elapsed between frames were irrelevant. What contributed to the radial speeds of the target and/or platform was how much motion was emulated in between frames. For the sake of simplicity a PRI of 4 milliseconds was assumed to help with the calculations. This yielded a Doppler sample range of -125 Hz to 125 Hz. A relatively slow moving target like a pedestrian walking at 0.5 m/s would then yield about 50 Hz Doppler frequency if the target were to be moving only radially with respect to radar. Assuming a 4ms PRI, it would mean a target that moves 0.5 m/s would have to move 2mm every PRI interval (between each frame). When the platform motion is emulated, it was decided to go for a speed as fast possible which was still within the Doppler sampling range to keep it simple and void Doppler aliasing. So with this setup it could only be a maximum 100 Hz with respect to ground-stationary targets (some margin was decided to be left from the limit of 125Hz as a rule of thumb). So when platform motion was emulated 1 m/s second speed was emulated which meant 4 mm of movement between each frame.

As a result a moving target with ~50Hz Doppler frequency was placed at around 4.35 m range with -3 degree azimuth angle. A stationary target was placed at 3.6 m range with -9 degree angle. The sketch of the complete scenario can be found in Figure 6-4 and the photo of the target scenario can be seen in Figure 6-5.

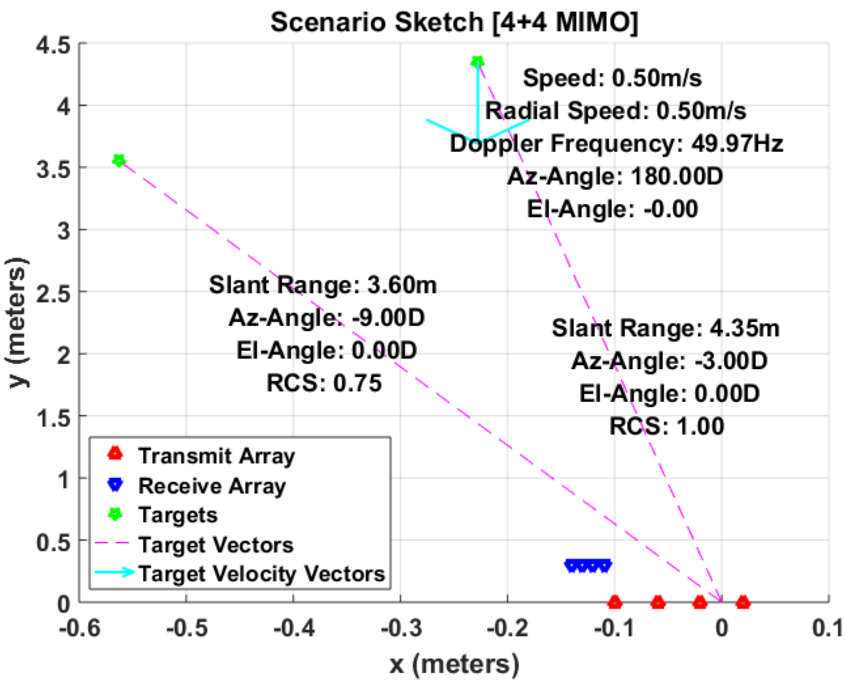


Figure 6-4 Experimental setup for GMTI with a 1D MIMO array (shown here with un-deviated array)

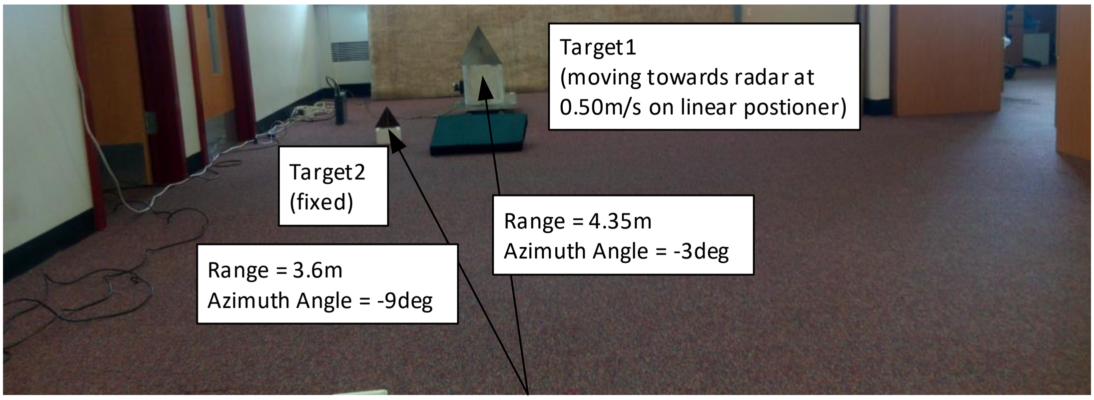


Figure 6-5 Target scenario for GMTI measurements.

Experimental Results

In this section the results from scaled environment experiments are presented. As far as the order of signal processing is concerned; first is beamforming on each frame yielding range-angle maps from each pulse. And second is GMTI processing applied to each range-angle point (each equivalent to a point in ground) through returned pulses. The result is a normalized return strength at possible range, angle and Doppler frequencies. Two ways to analyse this data is by drawing range-angle maps at specified Doppler frequencies or drawing range-Doppler maps at specified angles. Another method is obtaining a generic range-Doppler map of the entire angular spectrum (namely MIMO MTI maps) and obtaining return beams from chosen range-Doppler bins. When the results are presented all three of these methods are used to explain the details in much detail as possible, this may not be necessary for practical purposes.

Both the relative motion of platform and the relative motion of targets were emulated in a stop-and-go manner. Emulated speeds were in terms of distance per pulse, which made the PRI value irrelevant. However, for processing purposes a value had to be chosen, and for this purpose 4 ms was chosen since it's also the original system parameter. The Doppler frequencies and radial speeds presented below are calculated assuming a PRI of 4ms.

Stationary Platform

In this experiment it was simply tested if a moving target could be measured with our experimental setup. The setup involves a stationary platform. A stationary target was place around 3.6 meters and around -9 degrees. A moving target was placed at 4.3 meters and at around -3 degrees. The target was moved with a fixed speed of 2mm/PRI in Y direction relative to the array(note that the array is lined up on the X axis), which corresponds to a speed of about 0.5 m/s and a Doppler frequency of about 50 Hz. The GMTI processing was done with 19 frames.

Below in Figure 6-6 is the Range-Doppler map obtained via a non-coherent sum of all the range-Doppler maps in all scan-angles. In Figure 6-6, it is clearly visible that there exists a target at around 50Hz and 4.4 meters range. Due to the high-pass filtering used to avoid clutter reflection, the stationary target is supressed. And due to the number of pulses sent the Doppler resolution is limited therefore it is impossible to look at exactly 50Hz where target is expected to be found, but it's in the nearest Doppler bin.

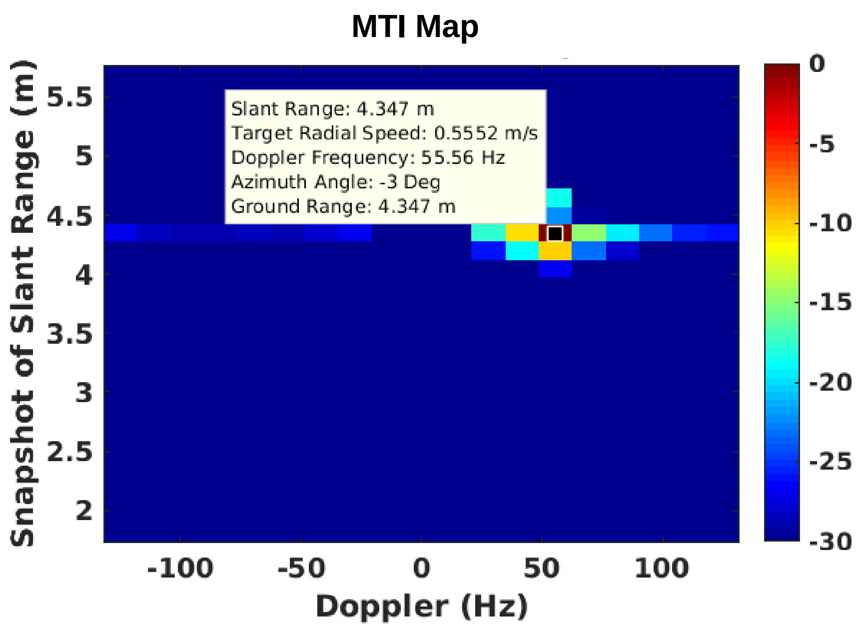


Figure 6-6 GMTI map (stationary platform and a single moving target)

In Figure 6-7 is the azimuth cut at 4.35 meters and 55Hz. At 4.35 meters, 55Hz and -3 degrees a strong return can be seen which corresponds to where our target is and its emulated Doppler frequency. The angular beamwidth and sidelobe levels are within expected range for a 4x4 MIMO array.

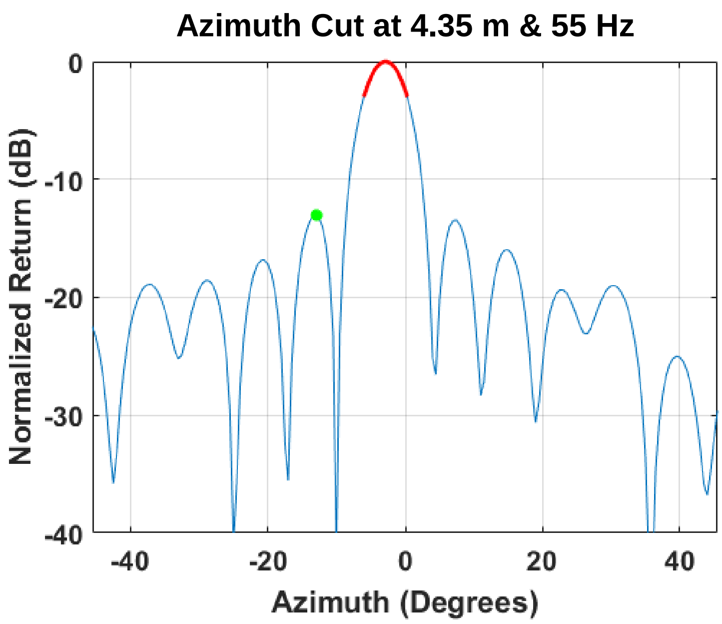


Figure 6-7 Azimuth cut at 4.35 meters and 55Hz (stationary platform)

<u>Moving Platform</u>

After verifying that our experimental setup works very well to emulate moving targets, next was emulating a moving platform and to test our platform motion compensation algorithm. All parameters were kept same apart from the moving platform. Platform was moved 4mm in X direction between each pulse, emulating a 2 m/s speed. In Figure 6-8 and Figure 6-9, the range-Doppler maps before and after platform motion compensation are presented. In Figure 6-8 even though the moving target can be seen roughly at where it is expected, other reflections can also be observed just around zero-Doppler band; this is because stationary targets have now a relative speed to the platform and therefore they start to show up.

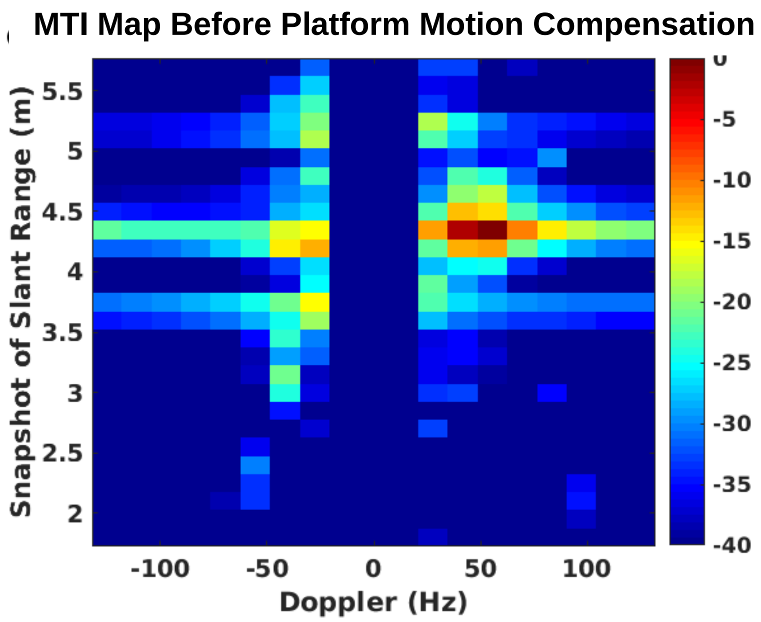


Figure 6-8 GMTI map before platform motion compensation (moving platform and a single moving target)

In Figure 6-9 below the stationary target can now observed to have shifted back to the zero Doppler band and shows a symmetrical structure (regarding Doppler sidelobes). The moving target can also be seen focused in a single Doppler bin which was occupying two Doppler bins before. In Figure 6-10 below it can also be observed that target can be located at -3 degrees.

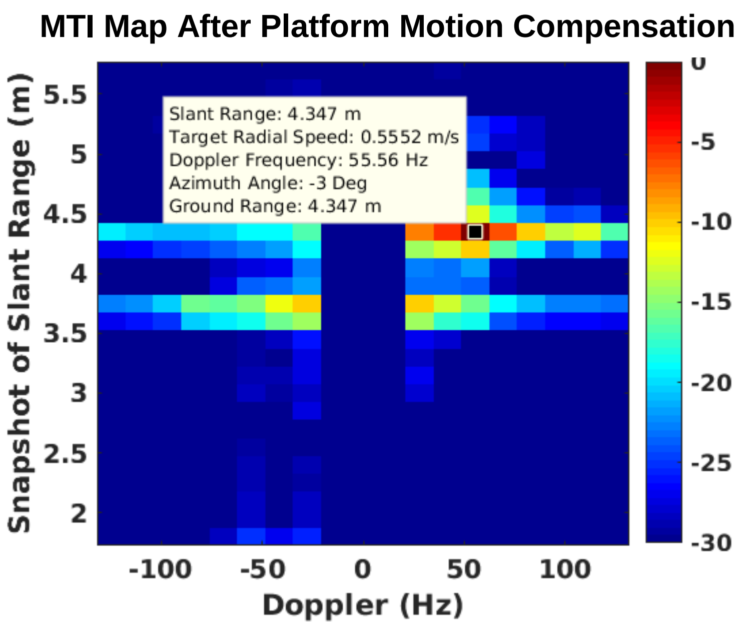


Figure 6-9 GMTI map after platform motion compensation (moving platform and a single moving target)

In Figure 6-10 is the azimuth cut at 4.35 meters and 55 Hz. It can be seen that the target can be localised at around -3 degrees where it was before. It can also be seen that the due to motion platform compensation the sidelobes have been reduced further. This happens because the original sidelobes which resided on the same range-Doppler plane now reside on a curved cone-like shape on 3 dimensions due to platform compensation. Therefore when the same plane is inspected, not the actual sidelobes are seen but their residues are observed. Nevertheless, this does not affect the localisation capabilities.

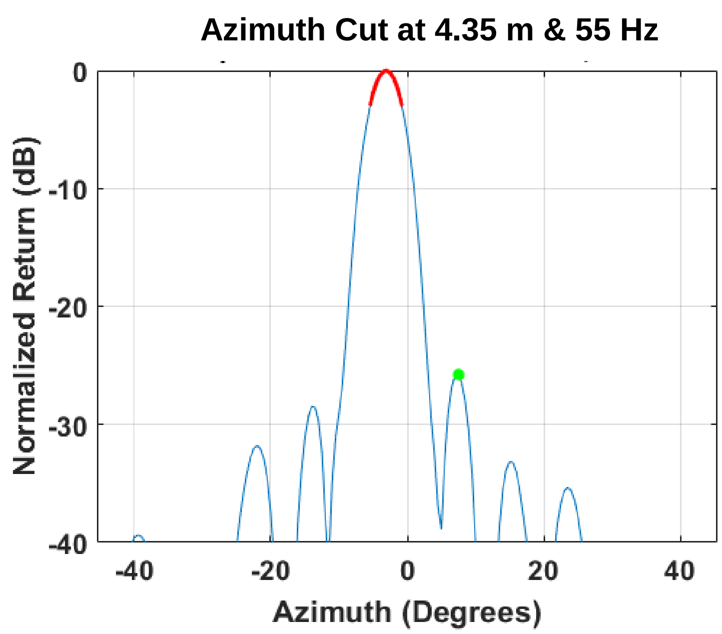


Figure 6-10 Azimuth cut at 4.35 meters and 55Hz (moving platform)

Simulation Results

In this section simulation results complementing scaled environment experiments are presented. The simulation program simulates waves sent and collected from the targets in a stop and go motion. It assumes point targets only and it doesn't compute multi-path reflections. In order to simulate VNA function a chirp waveform was assumed. The return signals FFT was taken and then it was fed to the same processing program as if the data was taken from the experimental setup. The simulation program replicated the scenarios created during the experiments. There are two targets; one stationary and one moving target. In Figure 6-11 the simulation scenario that's used can be observed.

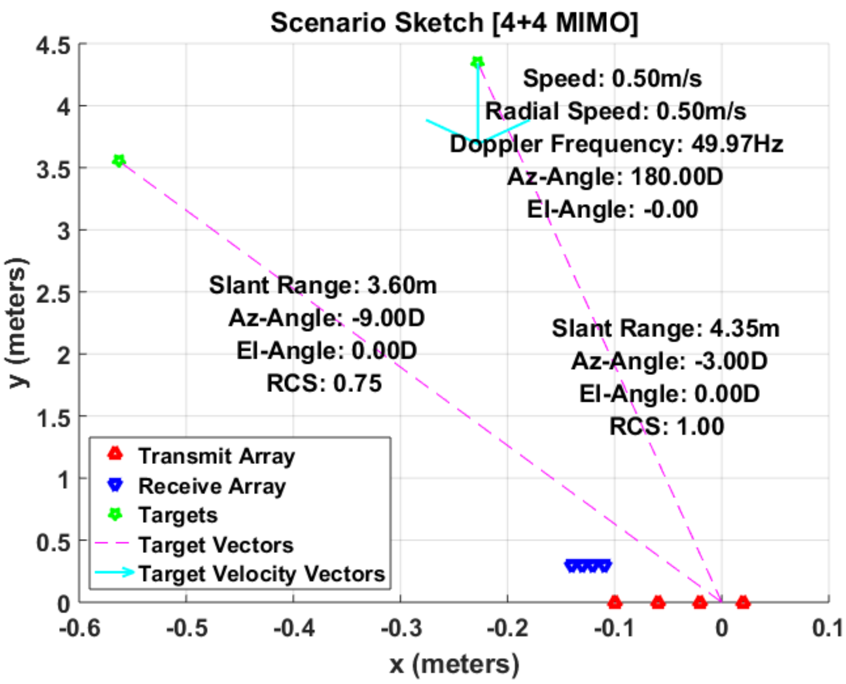


Figure 6-11 Scenario sketch for simulations

<u>*Stationary Platform*</u>

First simulation is simply a simulation of a MIMO array on a stationary platform looking at one moving target at 4.35 meters at around -3 degrees, and one stationary target at 3.6 meters at around -9 degrees. To match the emulated speed in experiments, the moving target was simulated with 0.5 m/s moving in the Y direction. In Figure 6-12 above it is clearly visible that our moving target can be observed at around 50Hz and 4.2 meters range. Due to the high-pass filtering used to avoid clutter reflection, the stationary target is supressed. Similar with the experimental results, due the number of simulated pulses results cannot exactly focus at 50Hz where the moving target is expected to be seen but rather on 55 Hz which is nearest Doppler bin.

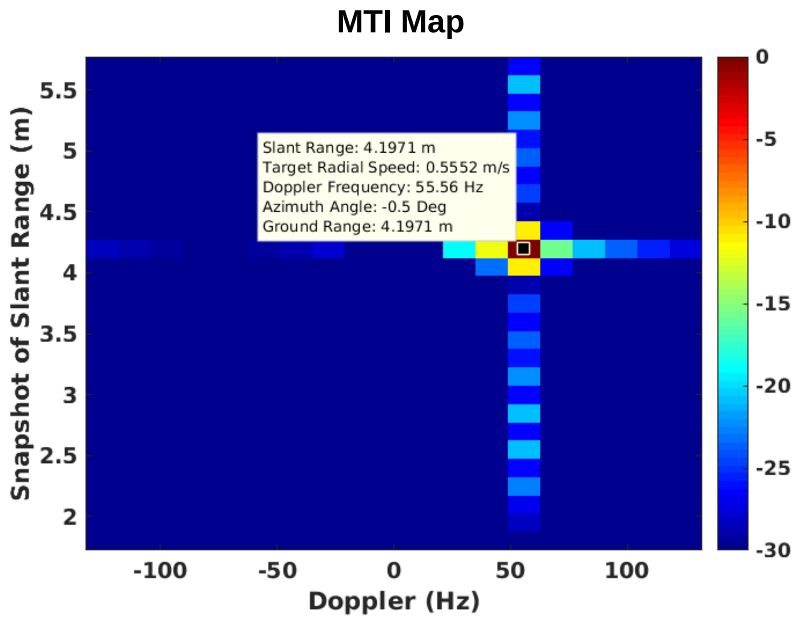


Figure 6-12 GMTI map from stationary platform and a single moving target

In Figure 6-13 below is the azimuth cut at 4.2 meters and 55Hz. At 4.2 meters, 55Hz and -3 degrees a strong return can be seen at where the moving target is and its emulated Doppler frequency. The angular beamwidth and sidelobe levels are within expected range for a 4x4 MIMO array.

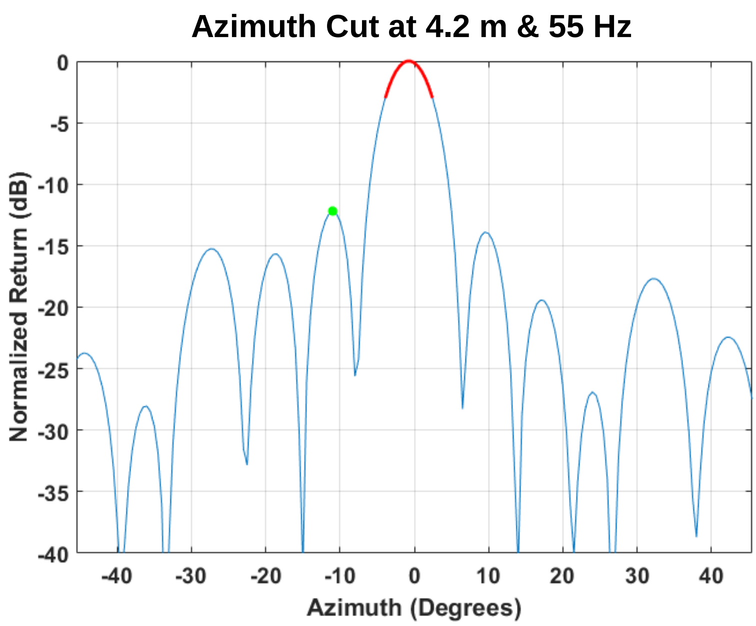


Figure 6-13 Azimuth cut at 4.2 meters and 55Hz (stationary platform)

Moving Platform

After verifying that our simulation results match experimental results with moving targets, next was simulating a moving platform. Like with the experimental setup, all parameters were kept same apart from the moving platform. Platform was simulated to have 2 m/s speed. In Figure 6-14 and Figure 6-15 below, the range-Doppler maps before and after platform motion compensation are presented. In Figure 6-14 even though the moving target can be seen roughly at where it is expected, other reflections just around the zero-Doppler band were also observed; this is because the stationary target have now a relative speed to the platform and therefore it starts to show up.

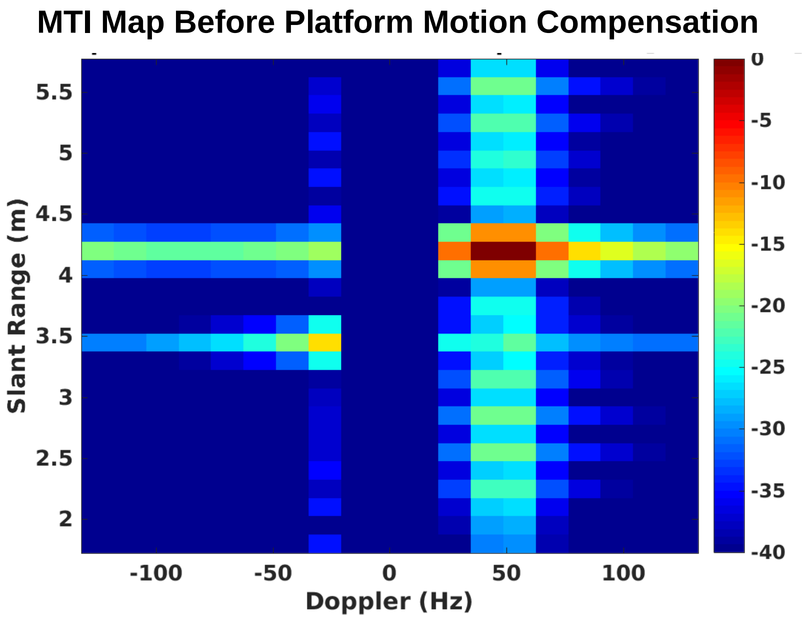


Figure 6-14 GMTI map before platform motion compensation (moving platform and a single moving target)

In Figure 6-15 below the stationary target can be seen to have shifted back to the zero Doppler band and shows a symmetrical structure (Doppler sidelobes). And also the moving target can now be seen focused in a single Doppler bin which was occupying two Doppler bins before, like in the experimental results. In Figure 6-16 below it can also be observed that target can be located at -2.5 degrees.

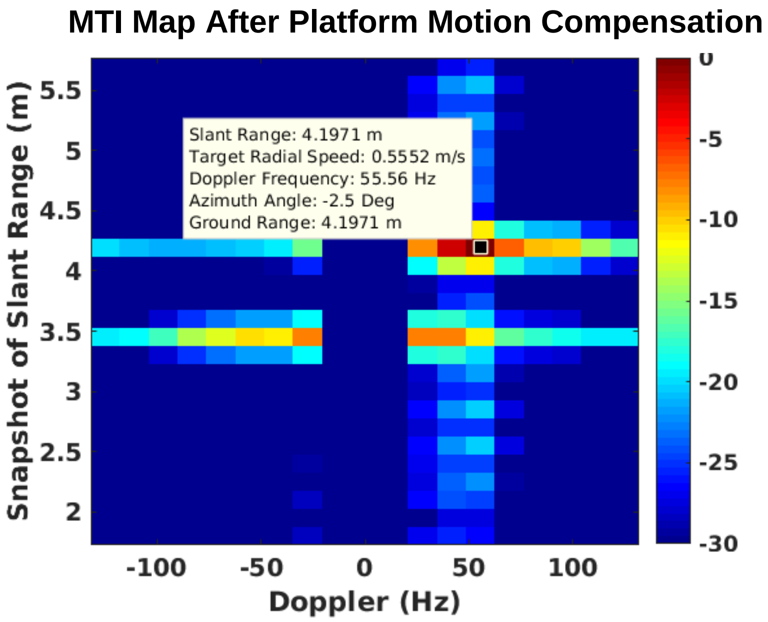


Figure 6-15 GMTI map after platform motion compensation (moving platform and a single moving target)

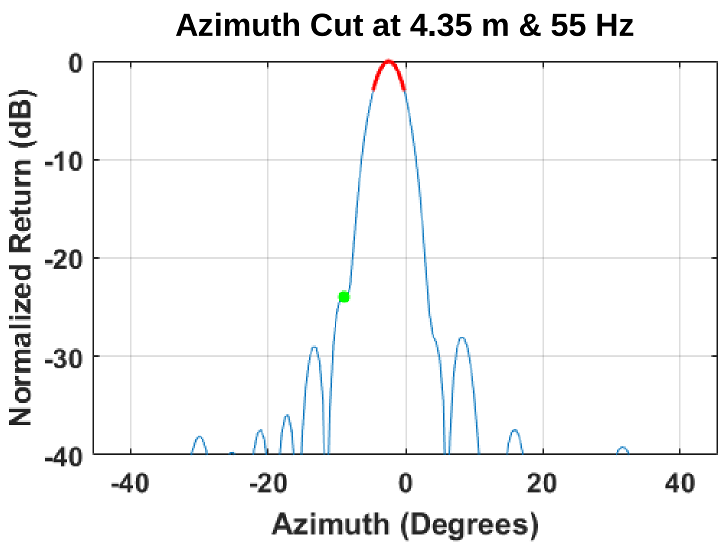


Figure 6-16 Azimuth cut at 4.35 meters and 55Hz (moving platform)

6.4.4 Experimental Setup in 435 MHz Band

For outdoor experimental measurements a 4 input, 4 output software defined radio setup was used. The realisation of the setup was made possible using 2 of the 2-input 2-output USRP (Universal Software defined Radio Peripheral) boxes in synchrony. The system was set to transmit and receive non-modulated pulses in TDMA scheme to obtain near-ideal orthogonality. The general system block diagram depicting the full setup can be seen in Figure 6-17.

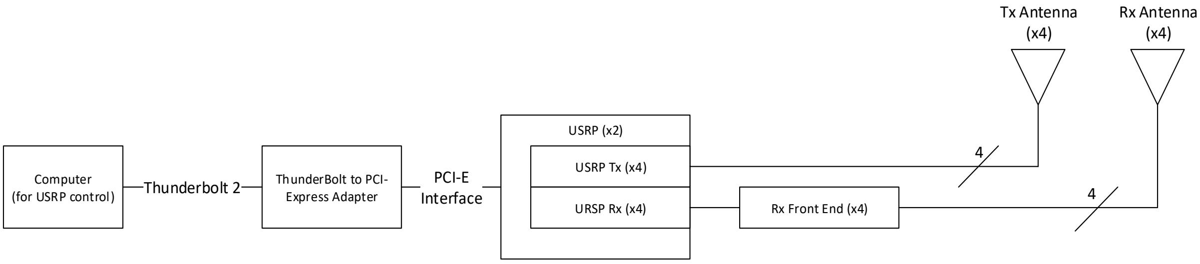


Figure 6-17 Setup for SIMITAR experiments at 435MHz.

The USRP setup was fully functional as a pulse MIMO radar with 8 narrowband dipole antennas with a centre frequency of 435 MHz (70cm band). All the antennas were physically placed and fixed on a fishing line as can be seen in Figure 6-18.

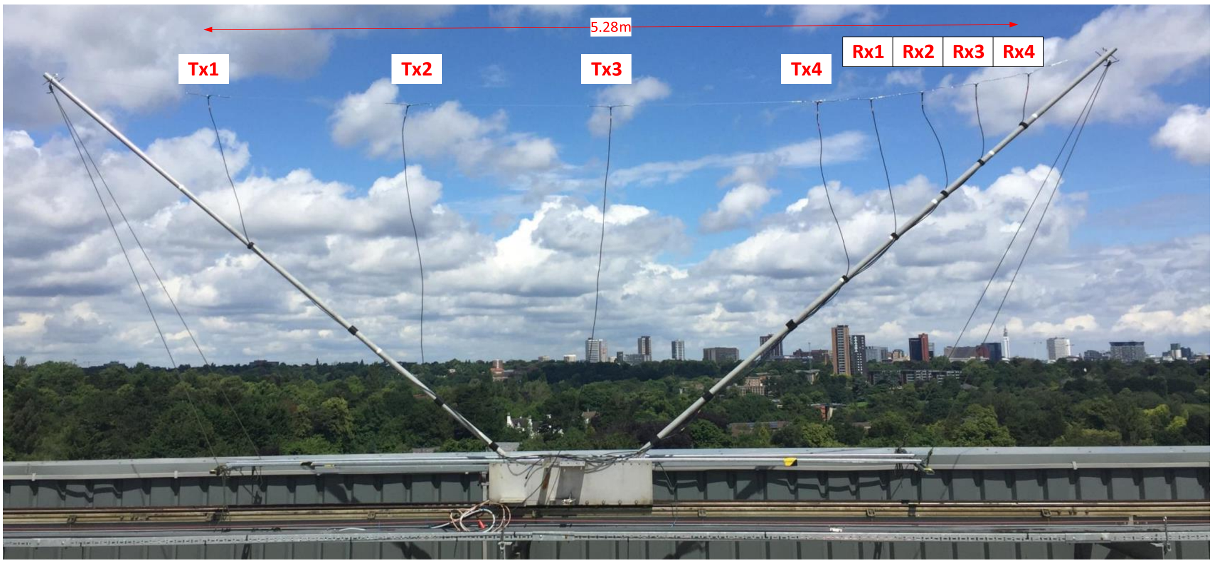


Figure 6-18 4Tx + 4Rx MIMO array used for MIMO experiments at 435MHz.

Receive array was implemented as the fine array with half-wavelength spacing and transmit array was placed next to it as the coarse array with double-wavelength spacing. The real distances that were used and the placement of the antennas can be observed in Figure 6-19.

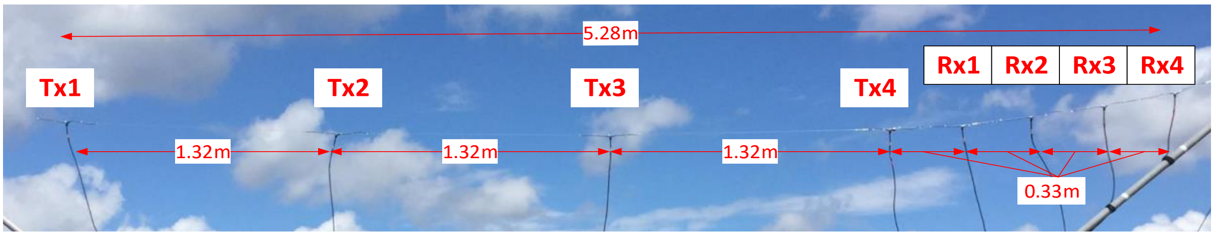


Figure 6-19 Spacing of antenna elements on the MIMO array.

The dipole antennas used were UHF 433 MHz semi-rigid dipoles, manufactured by IBCrazy, intended for use on consumer drones. A photo of the antenna can be seen in Figure 6-20.

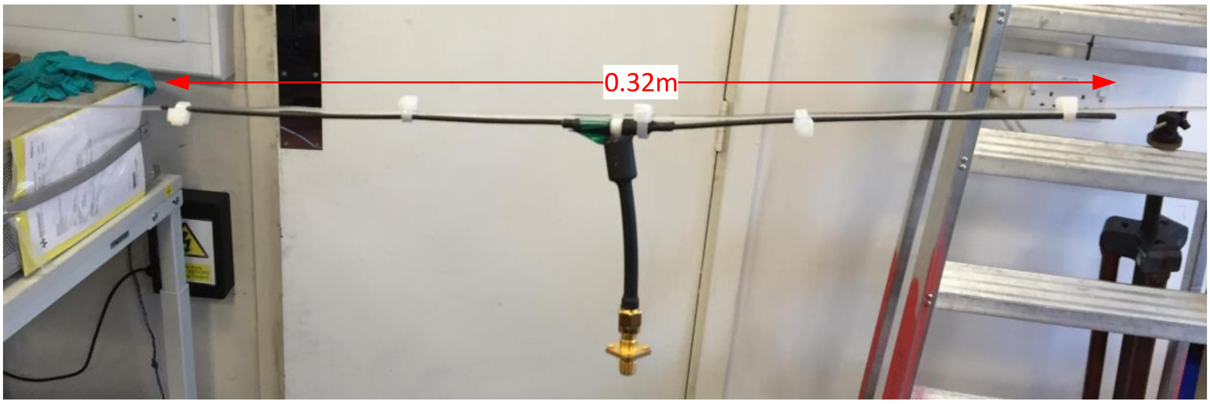


Figure 6-20 Antenna used on MIMO array for experiments at 435MHz.

Before using these antennas for experimental purposes certain tests were performed. Such as, within the intended operating frequency the self-inductance of the antenna was measured to be below -12dB. The S11 plot can be seen in Figure 6-21.

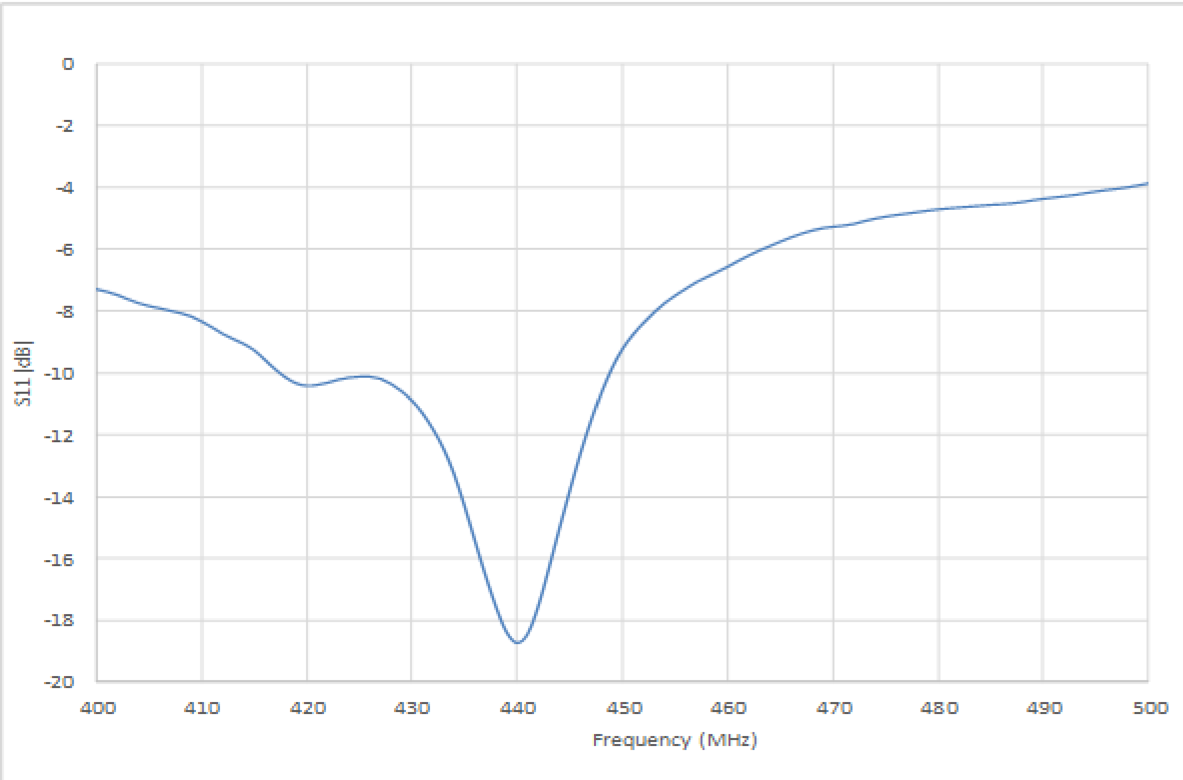


Figure 6-21 S11 response of a typical 433MHz antenna used for the SIMITAR experiment at 435MHz.

In addition, the mutual induction between closest pair of transmit and receive antennas were also measured (half-wavelength distance) and found to be better than 12 dB. The S21 plot can be seen in Figure 6-22.

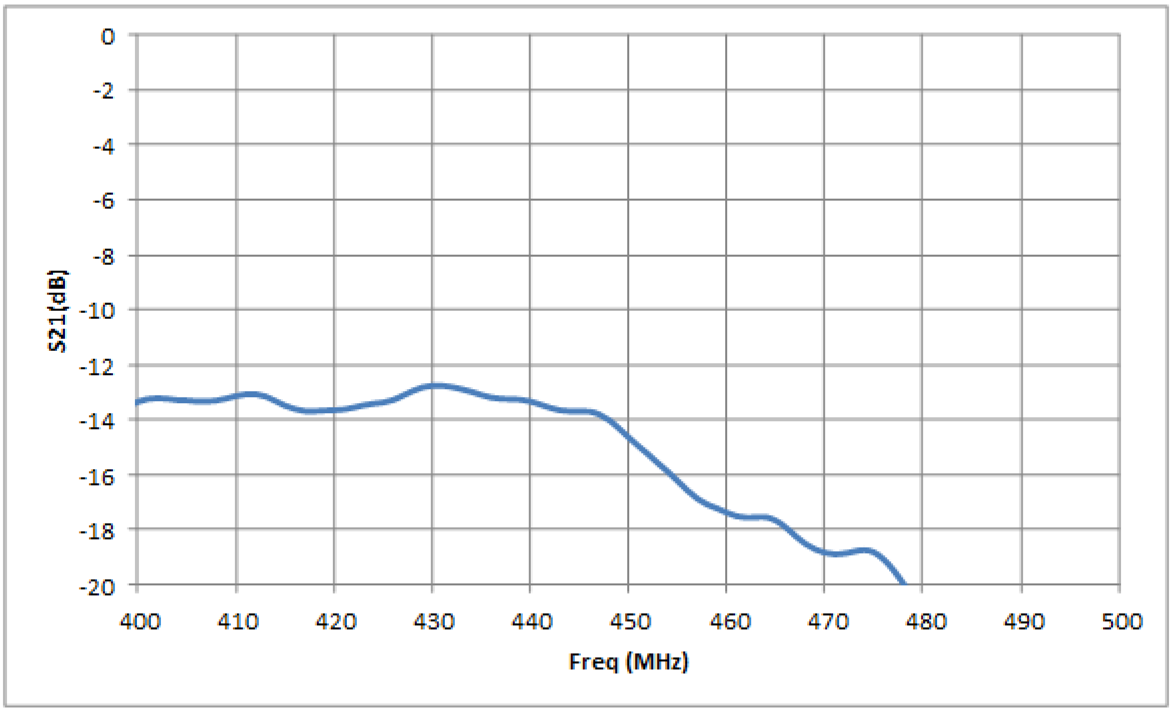


Figure 6-22 Measured S21 between closest pair of Tx/Rx antennas (Rx1 and Tx4) on array.

A relatively simple circuitry was used for the receiver front end. Since the USRP boxes already include up-converting, down-converting and filtering circuitry, only parts needed were attenuators and power limiters to protect the inputs from direct path transmissions. The system block diagram of the front end can be seen in Figure 6-23.

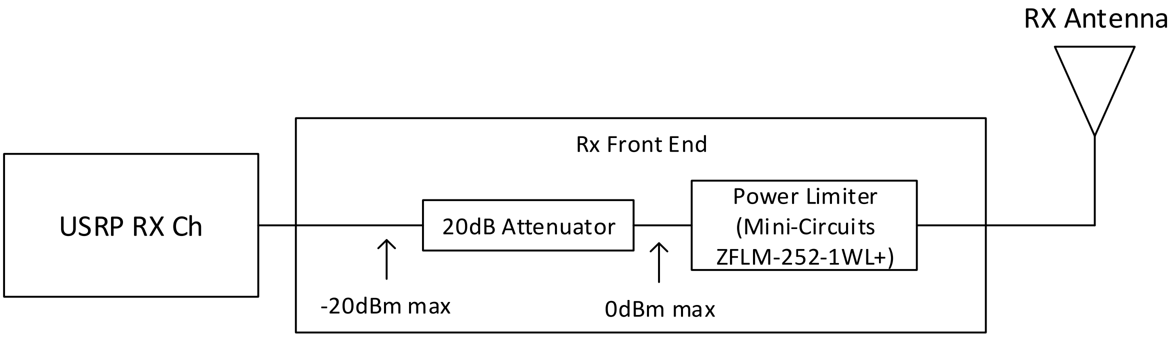


Figure 6-23 Rx Front End for each Rx channel

The power limiter was needed to limit the direct path transmissions to avoid burning out the inputs and also to avoid desensitising the receive channels, allowing targets to be detected. Output of the power limiter was 0 dBm. After the power limiter a 20dB attenuator was used to bring down the power levels even further to required level of maximum -15 dBm. For the implementation of the front end, off-the-shelf components from Micro-Circuits were used. A photo of the front end can be seen in Figure 6-24.

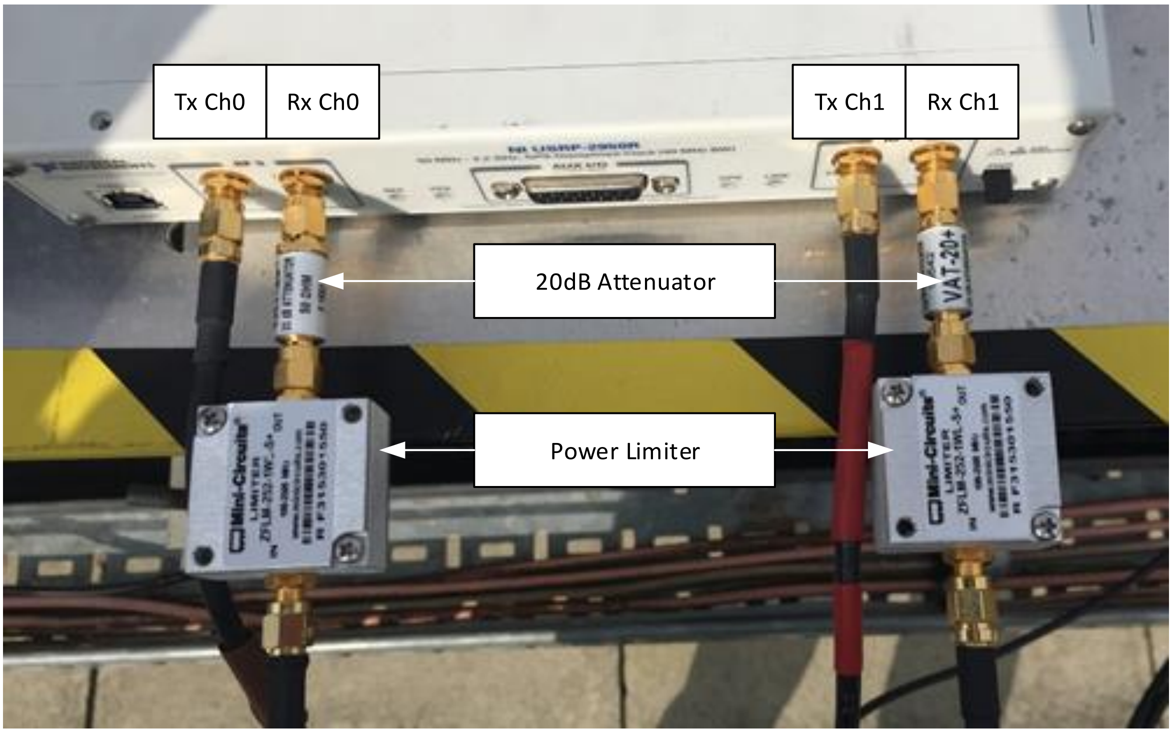


Figure 6-24 USRP Rx Front-end connection between antenna and USRP.

The USRP boxes used are software defined radio units with model number USRP-2950R from National Instruments. The boxes can and do perform the tasks of up-conversion of transmit signals, and filtering and down-conversion of received signals. Their main advantage is coherent and fast complex sampling of input and output signals. They can acquire and output baseband signals in real time and can even be programmed do certain signal processing operations on-board (this functionality was not used). A summary of specifications of the boxes can be found Table 6-2, where each channel can perform independently from each other. When configuring, only parameter that has to stay same along the channels is the sample rate and the trigger.

Table 6-2 Important parameters of the NI USRP-2950R

Parameter	**Value**	**Unit**
Frequency Band	50 to 2200	MHz
Maximum Tx Output power per port	20	dBm
Maximum Rx input power per port	-15	dBm
Number of Tx channels per USRP	2	-
Number of Rx channels per USRP	2	-

Rx Bandwidth	40	MHz

Since a single box has only 2 inputs and 2 outputs, two identical USRPs were used and connected together for coherent operation. A 10MHz reference signal and a PPS (pulse per second) trigger were used to ensure USRPs were phase coherent and timing clocks remain synchronised. A simple connection diagram can be seen in Figure 6-25.

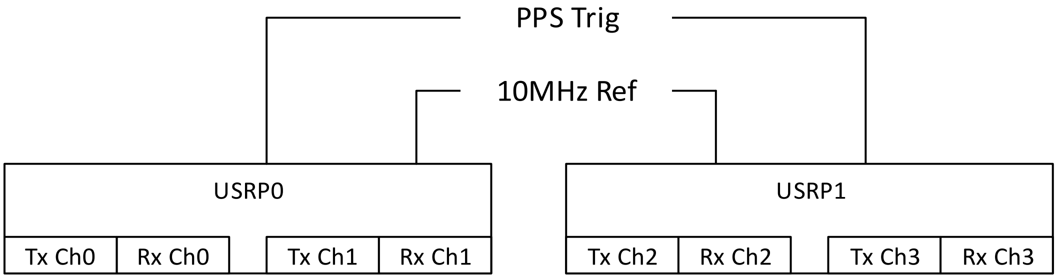


Figure 6-25 Connection between two USRPs for the experiment.

During the initial tests it was found the independency of channels of USRPs were more than expected. That is, the channels were found out to have their own local oscillators with different phases. So although, the channels were phase coherent, they were not necessarily phase aligned. This problem was solved by performing closed loop phase calibration of all channels pre-trials.

The system was built using a laptop rather than a computer with a tower in favour of mobility of the setup. This in the end limited the input data rate to 10 Megasamples/sec for various reasons. For the experiments this value dictated the signal bandwidth to be used, which was set to be 4MHz. As mentioned before a non-modulated pulse was used as the transmit waveform. This was found to be the most feasible option because modulated pulses were introducing too high sidelobe floors due to direct path signals and close-by stationary targets. It should be again noted that this experimental system was not necessarily optimised for automotive scenarios, where near-field measurements have utmost importance. A bandwidth of 4 MHz corresponds to a 0.25 us non-modulated pulse, and therefore a range resolution of 37.5 metres without any range windowing. The time between each frame –as was with the indoor setup- was set to 4ms. Inside the TDMA scheme however, the time between each pulse was set to allow 900 metres of range (two-way propagation), which corresponds to 6 us. To leave enough time between pulses and not to overload the computing units a 100 us inter-pulse duration was set. The timing summary of the transmission scheme can be seen in Figure 6-26.

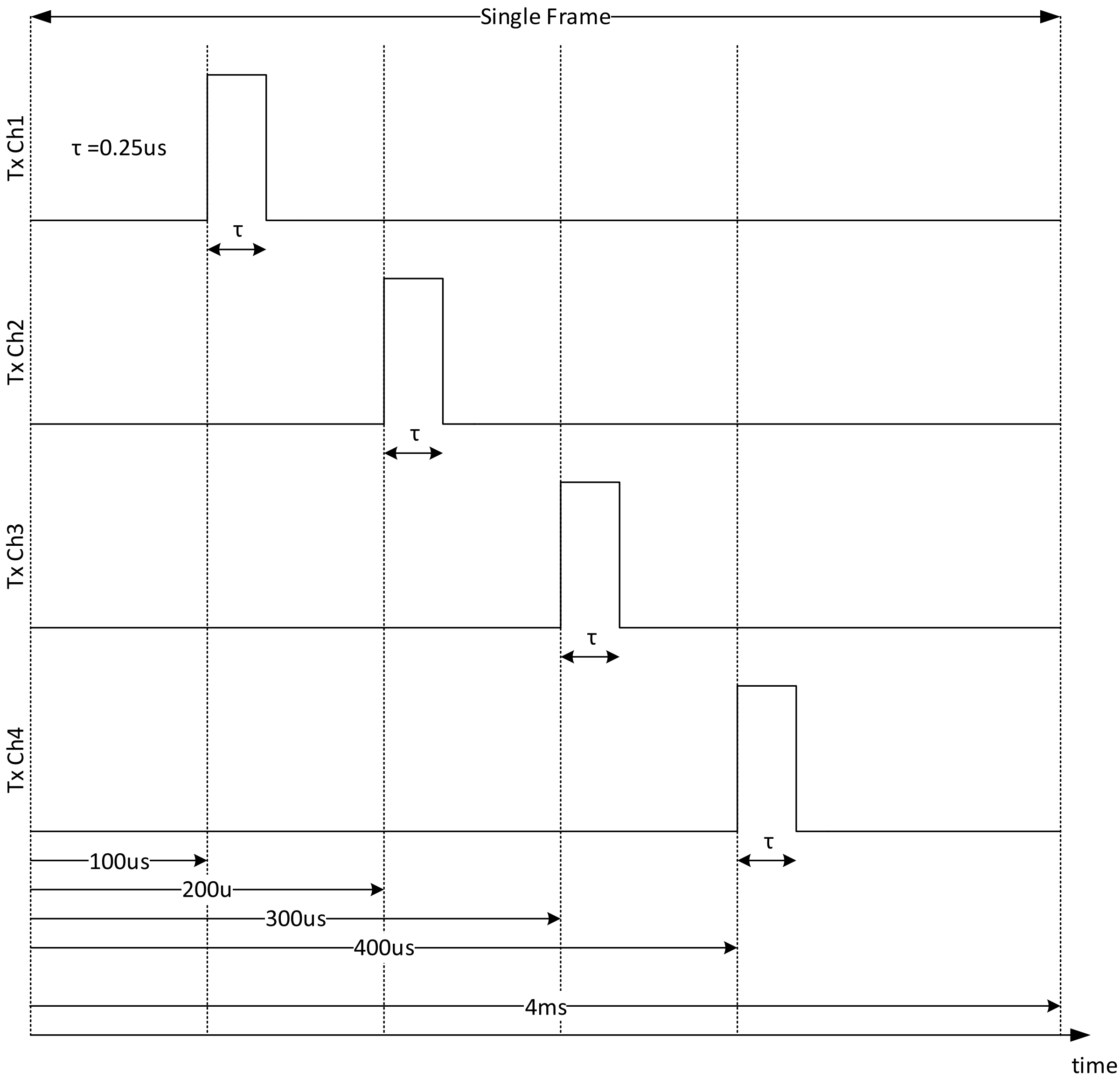


Figure 6-26 Specification of the signals sent to the Tx channels for each frame.

Since the transmit waveforms were sent out in TDMA scheme, it was already known when to expect which transmit signal. Therefore it was fairly trivial to trigger receivers and collect reflected signals that is simply cropping out received signals in 4 ms chunks. However, only the first 1 ms of data were recorded and saved after transmission for the sake of storage space and computing load. The timing summary of the reception can be seen in Figure 6-27.

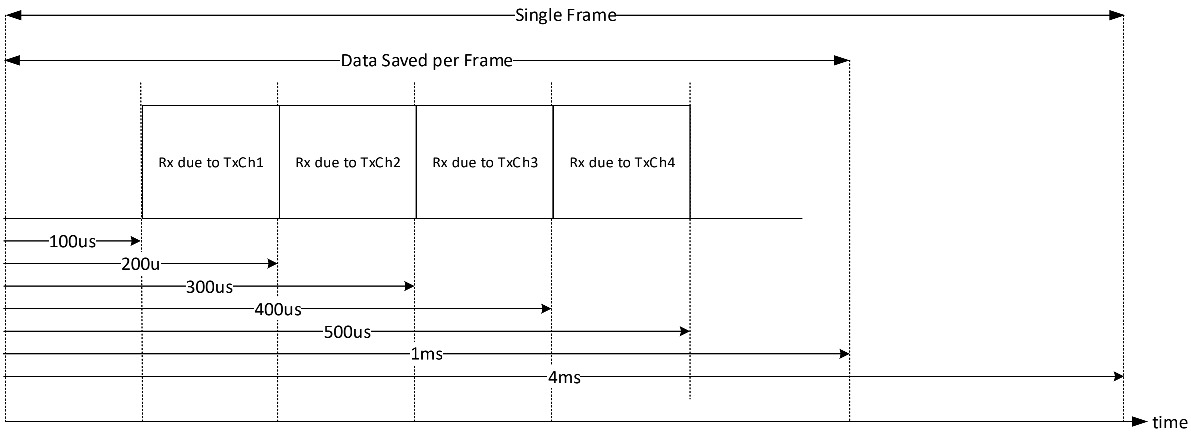


Figure 6-27 Structure of received data at each RX channel for each frame.

Finally the parameters used for this experiment are given in Table 6-3. The setup was located on the roof of Gisbert Kapp Building, University of Birmingham, Birmingham, where the antenna array was at an altitude of about 35m and looking at North.

Table 6-3 Parameters used for the SIMITAR experiments at 435MHz

Parameter	**Value**	**Units**
Number of Tx	4	-
Number of Rx	4	-
Carrier Medium	RF	-
Centre Frequency	435	MHz
Wavelength	70	cm
Waveform	Non-modulated Pulse	-
Bandwidth	4	MHz
Tx Element Spacing	1.32	m
Rx Element Spacing	0.33	m
Fraunhofer Range	>80	m
Pulse Width	0.25	us
Tx Power	19	dBm
PRI	4	ms
Complex Sample Rate	10	Megasamples/s

Altitude	35	m

Experimental Results

Rooftop experiments were done on a target availability basis. It wasn't up to us to choose target locations or their velocity.

Single Moving Target

First set of results is from observing a cement truck move towards us with an angle. In Figure 6-28 the initial position of the truck can be observed. And in Figure 6-29 below the final position of the truck can be seen after 6 seconds.

Figure 6-28 Initial position of the cement truck at the start of the measurement

Figure 6-29 Final position of the cement truck at the end of the measurement (6s)

In Figure 6-30 below the zoomed in MTI map looking at the range where the cement truck is located can be observed. The truck is clearly visible at around 100 meters range and about 2.5Hz Doppler frequency.

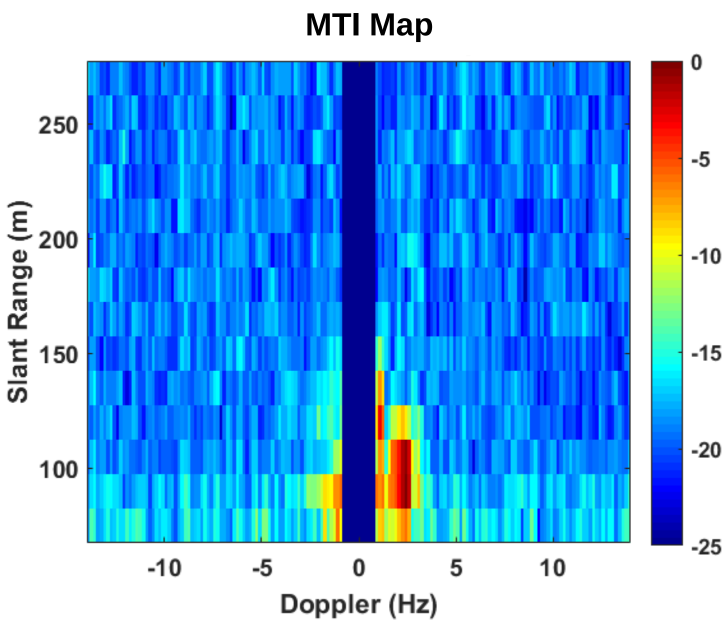


Figure 6-30 GMTI Map obtained from rooftop experiment observing a cement truck moving

Two Moving Targets in Opposite Directions

Second result that is with two trucks very close to each other moving at different directions. In Figure 6-31 the initial positions of the trucks can be observed. And in Figure 6-32 below the final position of the trucks can be seen after 6 seconds.

Figure 6-31 Initial positions of the trucks at the start of the measurement

Figure 6-32 Final positions of the trucks at the end of the measurement (6s)

In Figure 6-33 the GMTI map obtained from this measurement can be seen. The trucks can be observed at around 110 and 120 meters range and with about -2.5Hz and 1.5Hz Doppler frequency.

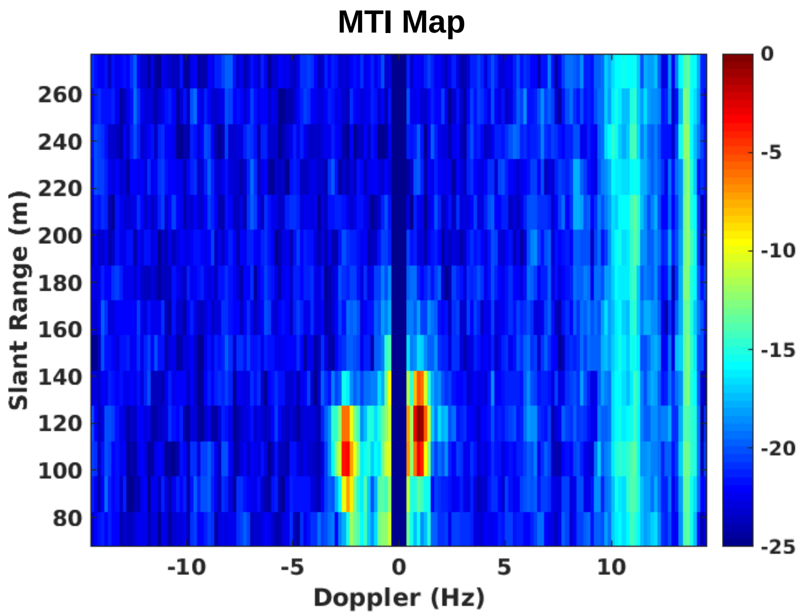


Figure 6-33 GMTI Map obtained from rooftop experiment observing two trucks moving in opposite directions

6.5 Conclusions

Two experimental setups were built and corresponding simulation programmes were developed to explore MTI applications with MIMO sensor arrays. Platform motion compensation algorithms were developed, tested and verified both experimentally and with simulations to encounter any Doppler shifts that may be caused by vehicular movement. Apart from indoor experiments, an outdoor experimental setup was used with targets of opportunity further verifying our findings. This chapter can be a pre-cursor study for a deeper inspection of possible MTI applications with MIMO sensor arrays

7 CONCLUSIONS & FUTURE WORK

7.1 CONCLUSIONS

The aim of this research was to investigate a novel Multiple-Input-Multiple-Output (MIMO) sensor system for automotive applications. Compared to traditional phased arrays, a MIMO array achieves the same fine angular resolution, but with a drastically reduced amount of sensor elements. For example, a MIMO array of 10 elements can deliver the same resolution as a phased array of 25 elements. The other highlight of this technology was that it can operate at short ranges, which is physically impossible with a phased array as the beam requires significant distance from the antenna aperture to form. Therefore a MIMO system could potentially provide very high angular resolution at short ranges. These properties make such a system attractive for a number of automotive applications, including parking aids, short-range cruise control, speed-over-ground estimation, pedestrian and object detection and collision detection.

Research started with the verification of application of conventional MIMO techniques for radar context. MIMO techniques were tested with the existing RF equipment in laboratory environment. Beamforming capabilities were verified, range and angular resolutions were compared to equivalent phased arrays and multiple-target resolving capabilities were confirmed. This first set of experiments was performed with 3 transmit elements and 5 receive elements with 3.5GHz carrier frequency. The experimental performance (i.e. beamwidth, sidelobe level, etc.) of the array was in par with a simulated 15 element phased array system.

Nearfield focusing algorithms for MIMO arrays were developed and verified via experiments in the anechoic chamber with the same equipment. Parallel to experiments and development, an extensive highly-configurable MIMO sensor array simulator program was developed to allow for testing of different technologies, signal processing algorithms, waveforms and non-conventional (aperiodic) MIMO configurations. Fraunhofer distance for an equivalent 15 element phased array is about 9 meters and Fraunhofer distance for a 5 element array (largest physical aperture in experimental setup) was 2 meters. Yet beamforming and target resolving capabilities with ranges as near as 1 meter were achieved experimentally. These results were compared to their simulation counterparts and were found to be in agreement.

A technology demonstrator based on ultrasonic equipment was built. It was decided to have 4 transmit elements and 4 receive elements and to have acoustic sensors to complement the RF setup since RF equipment was ready. An acoustic MIMO sensor array was designed and built as technology demonstrator. This technology demonstrator was built with modularity in mind to allow for upgrades and/or changes. In order to achieve this goal, transmit and receive modules are designed as separate boards from each other and a separate housing for these boards was built, so that the element spacings could be adjusted as easy as building a new housing. The technology demonstrator was then tested in an anechoic chamber and the findings were compared to computed performance parameters. The results were in agreement with expected parameters.

Further performance optimisations via aperiodic MIMO configurations were explored via use of heuristic optimisation algorithms. Three of the well-known and commonly used algorithms were modified to address our constraints and performance variables. A configuration obtained via heuristics was tested via simulations and a housing for the sensor array was 3D printed. This configuration then was tested in anechoic chamber and its performance was confirmed in practice. The aperiodic MIMO configuration that we have tested has about 4.25 degrees beamwidth only with 8 elements, with around -7.3dB sidelobe levels. This is equivalent to 35% improvement with respect to a conventional 8 element MIMO array or a 16 element phased array, and it has the equivalent beamwidth of a 24 element phased array. This design was experimentally tested and confirmed simulated performance. In addition, CLEAN algorithm was implemented to compensate for the overall higher sidelobe levels that can be expected in thinned arrays.

Finally using radio equipment again, an initial study on MTI applications was done. Using an indoor and an outdoor setup, MIMO MTI methods were developed and tested using stationary platforms. Further on, platform motion compensation methods were developed and tested in order to make up for vehicular motion and to compensate for its possible effects. The indoor setup was also used to emulate a moving platform and also used to experimentally confirm the MTI capabilities on a moving platform.

From an industrial point of view, the results of this thesis should make it relatively easy to implement a real-time MIMO radar for an automobile. Most -if not all- the components for our technology demonstrator was built from off the shelf products. For measurements, TDMA was used partly due to lack of smaller sensors in the market at the time. However, this is not an issue anymore, and a MIMO system capable of simultaneous transmission can be implemented also purely with off-the-shelf components. If there are concerns about signal orthogonality about using other multiple access schemes, the system could still use TDMA. Especially if ultrasonic were to be used, with development of custom electronics and on-board real-time processing and with proper coherent signal measurement; size, response time, electrical power and processing power requirements can be lowered to much less than what our technology demonstrator offers. Note that these parameters were not measured as it was not in the scope of this research, and this suggestion is based on back-of-the envelope calculations and estimates. Also, given that most of the tests were performed in an anechoic chamber an exploration of additional processing methods might be necessary to filter out clutter and interference that may become an issue in real life applications. Even then, there seems to be no obvious reason preventing the development of MIMO sensor applications in automotive context, especially in detection, ranging and scanning applications.

7.2 FUTURE WORK

There is always room for improvement at the end of each research. There are at least three key points that could be addressed as part of continuation of this project.

First point is about testing and verification of different orthogonality schemes. In order to achieve perfect orthogonality and also due to hardware limitations TDMA was used. At the time of the design and implementation of the acoustic technology demonstrator, there were no off-the-shelf ultrasonic transducers with small enough physical footprint to possibly fit into a half-wavelength spacing in stock. Once available, the SiSonic transducers from Knowles Electronics (or similar) could be used to implement a full physical MIMO array that could employ a different orthogonality scheme [94]. The proper orthogonality scheme and compensation for each drawback it introduces such as loss of sensitivity, direct signal compensation too are areas to be considered.

Second point is about further experimental validation of the aperiodic MIMO configurations. The feasibility of generating aperiodic MIMO configurations and beam pattern synthesis has been simulated and verified in laboratory conditions. It's even been taken one step further and tested with a practical object (a mountain bicycle). However, the testing of the technology demonstrator (or similar technologies) outside the laboratory conditions can open the way for further research by identifying possible real-life problems

Third point is about exploring a non-stochastic optimisation method for finding aperiodic MIMO array configurations. The heuristic methods that were used in this research have yielded great usable configurations for us. However, further study of aperiodic (or minimum-redundancy) MIMO arrays and a breakdown of what leads to a desirable beam pattern is needed. With that study, a deterministic method -as done in literature for other types of arrays- would be equally valuable for the further optimisation of MIMO sensor arrays and MIMO sensing field itself.

7.3 PUBLICATIONS AND DEMONSTRATIONS

No publications were made during this research up until the submission of the thesis. There are currently 1 journal paper submitted and 1 conference paper submitted and accepted involving the work in this thesis. There is another journal paper in the writing as of the time of submission of this thesis. There were also 2 attendances to poster conferences. Below is a list of publications and events in a chronological order.

- Poster conference attendance to 2016 Jaguar Land Rover PhD / EngDoc Conference on 1 December 2016. Awarded the second runner up in the competition.
- Poster conference attendance to EMSIG Radar Away Day in Crowne Plaza on 2 June 2017. Won the poster competition.

- "MIMO radar concept with a towed antenna array" by S. Pooni, A. Sayin, A. Stove, M. Antoniou, M. Cherniakov. Submitted to IET Belfast International Conference on Radar Systems on 29 June 2017.
- "MIMO Array for Short-Range, High-Resolution Automotive Sensing" by Michael Antoniou, Alp Sayin, Edward George Hoare, Sukhjit Pooni, Submitted to IET Radar Sonar & Navigation Automotive Special Issue on 9 March 2018.
- "Aperiodic MIMO Array for Short-Range, High Resolution Sensing for Robotics", Alp Sayin, Edward George Hoare, Sukhjit Pooni. Work in progress.

7.4 CONTRIBUTIONS

During this research, other people's and entities' contributions made it possible this research outcome. Below is a list of the people and entities contributed to this work and how.

- Development of initial experimental setup used in chapters 2 and 3 involved a team consisting of Daniele Stagliano, Stanislav Hristov, Sukhjit Pooni and the author of this thesis.
- The custom-built omnidirectional antennas used in this experimental setup were built by Edward Hoare.
- The electronic circuit design of transmitter and receiver modules which is the part of technology demonstrator was also done by Edward Hoare and their PCB design was done by Khairul Khaizi Mohd Shariff.
- When the work with DAQ board was started, initial guidance on how to use it was given by Liam Daniel.
- SIMITAR project's hardware development was done by Sukhjit Pooni and the project was funded by DSTL.
- This research was jointly funded by Jaguar Land Rover and University of Birmingham School of Electronic, Electrical and Systems Engineering.

REFERENCES

[1] M. A. Richards, J. Scheer, and W. A. Holm, *Principles of modern radar*. Raleigh, NC: SciTech Pub., 2010.

[2] H. Griffiths, C. Baker, D. Adamy, and G. W. Stimson, *Stimson's introduction to airborne radar*, Third edition. Edison, NJ: SciTech Publishing, 2014.

[3] M. I. Skolnik, *Introduction to radar systems*, 3. ed., [Nachdr.]. Boston, Mass.: McGraw Hill, 2001.

[4] P. Z. Peebles, *Radar principles*. New York: Wiley, 1998.

[5] M. Cherniakov and D. V. Nezlin, *Bistatic radar: principles and practice*. Southern Gate (Chichester): John Wiley & Sons, 2007.

[6] N. J. Willis, *Bistatic radar*. Edison, NJ: SciTech Publishing, 2005.

[7] M. Antoniou, Z. Zeng, L. Feifeng, and M. Cherniakov, 'Experimental Demonstration of Passive BSAR Imaging Using Navigation Satellites and a Fixed Receiver', *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 3, pp. 477–481, May 2012 [Online]. Available: http://ieeexplore.ieee.org/document/6086707/. [Accessed: 31-Aug-2017]

[8] F. E. Nathanson, J. P. Reilly, and M. Cohen, *Radar design principles: signal processing and the environment*, 2nd ed. New York: McGraw-Hill, 1991.

[9] M. E. Davis, *Foliage penetration radar: detection and characterization of objects under trees*. Raleigh, NC: SciTech Pub, 2011.

[10] M. T. Ma, *Theory and application of antenna arrays*. New York: Wiley, 1974.

[11] N. Fourikis, *Phased array-based systems and applications*. New York: Wiley, 1997.

[12] R. J. Mailloux, *Phased array antenna handbook*, 2nd ed. Boston: Artech House, 2005.

[13] C. A. Balanis, *Antenna theory: analysis and design*, 3rd ed. Hoboken, NJ: John Wiley, 2005.

[14] R. C. Hansen, *Phased array antennas*, 2nd ed. Hoboken, N.J: Wiley, 2009.

[15] H. J. Visser, *Array and phased array antenna basics*. Chichester, West Sussex, England ; Hoboken, NJ: John Wiley & Sons, 2005.

[16] R. Tulpule, N. Karnik, B. P. Kumar, and G. R. Branner, 'Near-field beamforming in conformal linear arrays', in *Antennas and Propagation Society International Symposium, 2002. IEEE*, 2002, vol. 2, pp. 120–123 vol.2.

[17] R. A. Kennedy, T. Abhayapala, D. B. Ward, and R. C. Williamson, 'Nearfield broadband frequency invariant beamforming', in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, 1996, vol. 2, pp. 905–908 [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=543268. [Accessed: 11-Aug-2014]

[18] R. A. Kennedy, D. B. Ward, and P. T. D. Abhayapala, 'Nearfield beamforming using nearfield/farfield reciprocity', in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, 1997, vol. 5, pp. 3741–3744 [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=604683. [Accessed: 11-Aug-2014]

[19] G. Qi-li and S. Chao, 'Time-domain nearfield wideband beamforming based on fractional delay filters', in *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on*, 2011, pp. 421–425 [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6013625. [Accessed: 11-Aug-2014]

[20] R. L. Haupt, J. J. Menozzi, and C. J. McCormack, 'Thinned arrays using genetic algorithms', in *Antennas and Propagation Society International Symposium, 1993. AP-S. Digest*, 1993, pp. 712–715 vol.2.

[21] A. Trucco and V. Murino, 'Stochastic optimization of linear sparse arrays', *Ocean. Eng. IEEE J. Of*, vol. 24, no. 3, pp. 291–299, 1999 [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=775291. [Accessed: 20-Apr-2015]

[22] M. A. Richards, *Fundamentals of radar signal processing*. New York: McGraw-Hill, 2005.

[23] M. A. Richards, J. A. Scheer, and W. A. Holm, *Principles of modern radar. vol. 2, vol. 2,*. Raleigh, N.C.: SciTech Pub., 2010.

[24] J. H. McClellan, *Signal processing first*. Upper Saddle River, NJ: Pearson/Prentice Hall, 2003.

[25] D. Bliss, K. Forsythe, and G. Fawcett, 'MIMO Radar: Resolution, Performance, and Waveforms', in *Proc. 14th Annual Adaptive Sensor Array Processing Workshop, MIT*, 2006, pp. 6–7.

[26] M. Lesturgie, 'Tutorial on MIMO Radar', Glasgow, UK, 21-Sep-2012.

[27] Y. Zhen, L. Wei, C. Qinggong, and H. Dahai, 'Design of a near-field radar imaging system based on MIMO array', in *Electronic Measurement & Instruments (ICEMI), 2015 12th IEEE International Conference on*, 2015, vol. 3, pp. 1265–1269 [Online]. Available: http://ieeexplore.ieee.org/abstract/document/7494513/. [Accessed: 04-Sep-2017]

[28] T. Spreng, U. Prechtel, B. Schonlinner, V. Ziegler, A. Meusling, and U. Siart, 'UWB near-field MIMO radar: Calibration, measurements and image reconstruction', in *Radar Conference (EuRAD), 2013 European*, 2013, pp. 33–36 [Online]. Available: http://ieeexplore.ieee.org/abstract/document/6689106/. [Accessed: 04-Sep-2017]

[29] X. Zhuge and A. G. Yarovoy, 'Three-Dimensional Near-Field MIMO Array Imaging Using Range Migration Techniques', *IEEE Trans. Image Process.*, vol. 21, no. 6, pp. 3026–3033, Jun. 2012 [Online]. Available: http://ieeexplore.ieee.org/document/6153063/. [Accessed: 04-Sep-2017]

[30] A. G. Yarovoy, T. G. Savelyev, P. J. Aubry, P. E. Lys, and L. P. Ligthart, 'UWB Array-Based Sensor for Near-Field Imaging', *IEEE Trans. Microw. Theory Tech.*, vol. 55, no. 6, pp. 1288–1295, Jun. 2007 [Online]. Available: http://ieeexplore.ieee.org/document/4230900/. [Accessed: 04-Sep-2017]

[31] L. Yu, S. Li, G. Zhao, and H. Sun, 'Near-field microwave imaging with a new optimization of MIMO array', in *Microwave Conference (APMC), 2015 Asia-Pacific*, 2015, vol. 1, pp. 1–3 [Online]. Available: http://ieeexplore.ieee.org/abstract/document/7411698/. [Accessed: 04-Sep-2017]

[32] U. Prechtel, V. Meenakshisundaram, B. Schoenlinner, V. Ziegler, H.-P. Feldle, and A. Meusling, 'Short-range MIMO radar system considerations', in *Antennas and Propagation (EUCAP), 2012 6th European Conference on*, 2012, pp. 1742–1745 [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6206599. [Accessed: 11-Aug-2014]

[33] T. Fromenteze, M. Boyarsky, J. Gollub, T. Sleasman, M. Imani, and D. R. Smith, 'Single-frequency near-field MIMO imaging', in *Antennas and Propagation (EUCAP), 2017 11th European Conference on*, 2017, pp. 1415–1418 [Online]. Available: http://ieeexplore.ieee.org/abstract/document/7928576/. [Accessed: 04-Sep-2017]

[34] F. G. Jansen, 'Automotive radar sensor for ultra short range applications', in *Radar Symposium (IRS), 2017 18th International*, 2017, pp. 1–6 [Online]. Available: http://ieeexplore.ieee.org/abstract/document/8008134/. [Accessed: 04-Sep-2017]

[35] B. Yang, A. Yarovoy, P. Aubry, and X. Zhuge, 'Experimental verification of 2D UWB MIMO antenna array for near-field imaging radar', in *Microwave Conference, 2009. EuMC 2009. European*, 2009, pp. 97–100 [Online]. Available: http://ieeexplore.ieee.org/abstract/document/5295977/. [Accessed: 04-Sep-2017]

[36] C.-Y. Chen and P. P. Vaidyanathan, 'Minimum redundancy MIMO radars', in *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*, 2008, pp. 45–48 [Online]. Available: http://ieeexplore.ieee.org/abstract/document/4541350/. [Accessed: 04-Sep-2017]

[37] A. Moffet, 'Minimum-redundancy linear arrays', *IEEE Trans. Antennas Propag.*, vol. 16, no. 2, pp. 172–175, 1968 [Online]. Available: http://ieeexplore.ieee.org/abstract/document/1139138/. [Accessed: 05-Sep-2017]

[38] Jian Dong, Qingxia Li, and Wei Guo, 'A Combinatorial Method for Antenna Array Design in Minimum Redundancy MIMO Radars', *IEEE Antennas Wirel. Propag. Lett.*, vol. 8, pp. 1150–1153, 2009 [Online]. Available: http://ieeexplore.ieee.org/document/5290079/. [Accessed: 04-Sep-2017]

[39] G. Liao, M. Jin, and J. Li, 'A two-step approach to construct minimum redundancy MIMO radars', in *Radar Conference-Surveillance for a Safer World, 2009. RADAR. International*, 2009, pp. 1–4 [Online]. Available: http://ieeexplore.ieee.org/abstract/document/5438446/. [Accessed: 04-Sep-2017]

[40] K. Rezer, W. Gropengie\s ser, and A. F. Jacob, 'Particle swarm optimization of minimum-redundancy MIMO arrays', in *Microwave Conference (GeMIC), 2011 German*, 2011, pp. 1–4

[Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5760699. [Accessed: 07-Dec-2014]

[41] A. Kirschner, U. Siart, J. Guetlein, and J. Detlefsen, 'A design-algorithm for MIMO radar antenna setups with minimum redundancy', in *Microwaves, Communications, Antennas and Electronics Systems (COMCAS), 2013 IEEE International Conference on*, 2013, pp. 1–5 [Online]. Available: http://ieeexplore.ieee.org/abstract/document/6685236/. [Accessed: 04-Sep-2017]

[42] J. Dong, F. Liu, Y. Jiang, B. Hu, and R. Shi, 'Minimum redundancy MIMO array design using cyclic permutation of perfect distance (CPPD)', in *General Assembly and Scientific Symposium (URSI GASS), 2014 XXXIth URSI*, 2014, pp. 1–4 [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6929489. [Accessed: 07-Dec-2014]

[43] J. Dong, R. Shi, Y. Guo, and W. Lei, 'Antenna array design in MIMO radar using cyclic difference sets and genetic algorithm', in *Antennas, Propagation & EM Theory (ISAPE), 2012 10th International Symposium on*, 2012, pp. 26–29 [Online]. Available: http://ieeexplore.ieee.org/abstract/document/6408693/. [Accessed: 04-Sep-2017]

[44] J. Dong, R. Shi, H. Ren, and D. Li, 'MIMO radar array thinning using almost difference sets', in *Antennas and Propagation (EuCAP), 2016 10th European Conference on*, 2016, pp. 1–2 [Online]. Available: http://ieeexplore.ieee.org/abstract/document/7481983/. [Accessed: 04-Sep-2017]

[45] J. He, D.-Z. Feng, N. H. Younan, and X. Li, 'Optimizing thinned antenna array geometry in MIMO radar systems using multiple genetic algorithm', in *Radar (Radar), 2011 IEEE CIE International Conference on*, 2011, vol. 1, pp. 971–974.

[46] X. Liu, C. Sun, Y. Yang, J. Zhuo, and Y. Han, 'Low complexity MIMO sonar imaging using a virtual sparse linear array', *J. Syst. Eng. Electron.*, vol. 27, no. 2, pp. 370–378, Apr. 2016 [Online]. Available: http://ieeexplore.ieee.org/document/7514425/. [Accessed: 04-Sep-2017]

[47] K. Rezer and A. F. Jacob, 'Thinned MIMO arrays with constrained element spacing for imaging radar', in *2012 9th European Radar Conference*, 2012, pp. 87–90.

[48] K. Rezer, C. Klickow, and A. F. Jacob, 'Thinned MIMO frame-arrays for radar imaging', in *Microwave Symposium Digest (MTT), 2012 IEEE MTT-S International*, 2012, pp. 1–3 [Online]. Available: http://ieeexplore.ieee.org/abstract/document/6259766/. [Accessed: 04-Sep-2017]

[49] 'Performance Analysis of Time Division and Code Division Waveforms in Co-Located MIMO'.

[50] 'MIMO Radar - Some Practical Considerations'.

[51] S. Wentao, H. Jianguo, C. Xiaodong, and H. Yunshan, 'Orthogonal waveforms design and performance analysis for MIMO sonar', in *Signal Processing (ICSP), 2010 IEEE 10th International Conference on*, 2010, pp. 2382–2385 [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5656922. [Accessed: 12-Feb-2014]

[52] Y. Zhang and J. Wang, 'OFDM-coded signals design for MIMO radar', in *Signal Processing, 2008. ICSP 2008. 9th International Conference on*, 2008, pp. 2442–2445 [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4697643. [Accessed: 04-Dec-2014]

[53] M. A. Richards, J. Scheer, W. A. Holm, and W. L. Melvin, Eds., *Principles of modern radar*. Raleigh, NC: SciTech Pub, 2010.

[54] W.-D. Wirth and Institution of Electrical Engineers, *Radar techniques using array antennas*. London: Institution of Electrical Engineers, 2001.

[55] Tektronix, 'AWG5000 and AWG7000 Series Arbitrary Waveform Generators Quick Start User Manual'. 30-Aug-2011 [Online]. Available: http://www.tek.com/awg7101-manual/awg7000-series-quick-start-user-manual. [Accessed: 19-Jul-2017]

[56] Tektronix, 'MSO70000C/DX Series Mixed Signal Oscilloscopes DPO70000C/DX Series Digital Phosphor Oscilloscopes DPO7000C Series Digital Phosphor Oscilloscopes MSO5000B and DPO5000B Series Oscilloscopes User Manual'. 29-Mar-2017 [Online]. Available: http://www.tek.com/oscilloscope/dpo70000-mso70000-manual-24. [Accessed: 19-Jul-2017]

[57] Hp/Agilent/Keysight, 'Agilent 8648A/B/C/D Signal Generators Data Sheet'. Keysight [Online]. Available: http://www.keysight.com/en/pd-1000002196%3Aepsg%3Apro-pn-8648D/synthesized-signal-generator-9-khz-to-4000-mhz?cc=GB&lc=eng. [Accessed: 17-Jul-2017]

[58] R. P. Feynman, R. B. Leighton, M. L. Sands, and R. P. Feynman, *Mainly mechanics, radiation, and heat*, Nachdr. Reading/Mass.: Addison-Wesley, 2007.

[59] L. Jakevičius and A. Demčenko, 'Ultrasound attenuation dependence on air temperature in closed chambers', *Ultragarsas Ultrasound*, vol. 63, no. 1, pp. 18–22, 2008 [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.216.4869&rep=rep1&type=pdf. [Accessed: 14-May-2015]
[60] H. E. Bass, L. C. Sutherland, A. J. Zuckerwar, D. T. Blackstock, and D. M. Hester, 'Atmospheric absorption of sound: Further developments', *J. Acoust. Soc. Am.*, vol. 97, no. 1, pp. 680–683, 1995 [Online]. Available: https://doi.org/10.1121/1.412989
[61] L. Jakevicius, A. Demcenko, and R. Mardosaite, 'Ultrasound attenuation dependence on air compression or expansion processes', *Ultragarsas*, vol. 65, no. 1, pp. 42–46, 2010 [Online]. Available: http://doc.utwente.nl/74546/1/ultrasound.pdf. [Accessed: 14-May-2015]
[62] Measurement Computing, 'USB-2537 User's Guide'. Measurement Computing Corporation, Jan-2011.
[63] M. Soumekh, *Synthetic aperture radar signal processing with MATLAB algorithms*. New York: J. Wiley, 1999.
[64] A. Newell and H. A. Simon, 'Computer science as empirical inquiry: Symbols and search', *Commun. ACM*, vol. 19, no. 3, pp. 113–126, 1976 [Online]. Available: http://dl.acm.org/citation.cfm?id=360022. [Accessed: 15-Apr-2015]
[65] B. Basu and G. K. Mahanti, 'Evolutionary algorithms for synthesis of uniform circular array with minimum side lobe level and maximum directivity', in *India Conference (INDICON), 2010 Annual IEEE*, 2010, pp. 1–4 [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5712620. [Accessed: 15-Apr-2015]
[66] C.-H. Hsu, 'Optimizing beam pattern of adaptive linear phase array antenna using local genetic algorithm', in *Antennas and Propagation Society International Symposium, 2005 IEEE*, 2005, vol. 1, pp. 315–318 [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1551553. [Accessed: 15-Apr-2015]
[67] D. S. Varma, R. Aditya, and K. R. Subhashini, 'Synthesis of adaptive antenna with circular geometry employing Harmony Search and Differential Evolution techniques', in *Communications and Signal Processing (ICCSP), 2013 International Conference on*, 2013, pp. 933–937 [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6577193. [Accessed: 15-Apr-2015]
[68] U. Singh and M. Rattan, 'Design of thinned concentric circular antenna arrays using firefly algorithm', *IET Microw. Antennas Propag.*, vol. 8, no. 12, pp. 894–900, Sep. 2014 [Online]. Available: http://digital-library.theiet.org/content/journals/10.1049/iet-map.2013.0695. [Accessed: 15-Apr-2015]
[69] R. L. Haupt, 'Thinned arrays using genetic algorithms', *Antennas Propag. IEEE Trans. On*, vol. 42, no. 7, pp. 993–999, 1994 [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=299602. [Accessed: 15-Apr-2015]
[70] D. J. O'Neill, 'Element placement in thinned arrays using genetic algorithms', in *OCEANS'94.'Oceans Engineering for Today's Technology and Tomorrow's Preservation.'Proceedings*, 1994, vol. 2, pp. II–301 [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=364059. [Accessed: 15-Apr-2015]
[71] M. G. Bray, D. H. Werner, D. W. Boeringer, and D. W. Machuga, 'Optimization of thinned aperiodic linear phased arrays using genetic algorithms to reduce grating lobes during scanning', *IEEE Trans. Antennas Propag.*, vol. 50, no. 12, pp. 1732–1742, Dec. 2002 [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1178722. [Accessed: 15-Apr-2015]
[72] K. Rezer and A. F. Jacob, 'Inst. fur Hochfrequenztech., Techn. Univ. Hamburg-Harburg, Hamburg, Germany', in *Radar Conference (EuRAD), 2012 9th European*, 2012, pp. 87–90 [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6450614. [Accessed: 07-Dec-2014]
[73] Chen, Chun-Yang, 'Signal processing algorithms for MIMO radar', Thesis (Dissertation (Ph.D.)), California Institute of Technology, 2009 [Online]. Available: http://resolver.caltech.edu/CaltechETD:etd-06082009-131045. [Accessed: 05-Jan-2014]

[74] N. Mladenović and P. Hansen, ‘Variable neighborhood search’, *Comput. Oper. Res.*, vol. 24, no. 11, pp. 1097–1100, 1997 [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0305054897000312

[75] P. Hansen, N. Mladenović, and J. A. Moreno Pérez, ‘Variable neighbourhood search: methods and applications’, *Ann. Oper. Res.*, vol. 175, no. 1, pp. 367–407, Mar. 2010 [Online]. Available: http://link.springer.com/10.1007/s10479-009-0657-6. [Accessed: 16-Apr-2015]

[76] S. Kirkpatrick, M. P. Vecchi, and others, ‘Optimization by simmulated annealing’, *science*, vol. 220, no. 4598, pp. 671–680, 1983 [Online]. Available: http://leonidzhukov.net/hse/2013/stochmod/papers/KirkpatrickGelattVecchi83.pdf. [Accessed: 20-Apr-2015]

[77] V. Černỳ, ‘Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm’, *J. Optim. Theory Appl.*, vol. 45, no. 1, pp. 41–51, 1985 [Online]. Available: http://link.springer.com/article/10.1007/BF00940812. [Accessed: 20-Apr-2015]

[78] Y. Yanli and W. Yingmin, ‘Beam Pattern Optimization Using MVDR and Simulated Annealing’, 2010, pp. 117–120 [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5569919. [Accessed: 20-Apr-2015]

[79] D. Adams, *The Restaurant at the end of the universe*. New York: Ballantine, 1995.

[80] A. M. Turing, ‘A quarterly review of psychology and philosophy’, *Pattern Recognit. Introd. - Dowden Hutchinson Ross Inc*, 1973.

[81] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*, 1st MIT Press ed. Cambridge, Mass: MIT Press, 1992.

[82] C. R. Reeves and J. E. Rowe, *Genetic algorithms: principles and perspectives: a guide to GA theory*. Boston: Kluwer Academic Publishers, 2003.

[83] R. Han, ‘Pattern Synthesis of Sparse Phased Array Antenna Using Genetic Algorithms’, *Mod. Appl. Sci.*, vol. 3, no. 9, p. p91, 2009 [Online]. Available: http://www.ccsenet.org/journal/index.php/mas/article/view/3809. [Accessed: 21-Apr-2015]

[84] B. Boudamouz, P. Millot, and C. Pichot, ‘MIMO antenna design with genetic algorithm for TTW radar imaging’, in *Radar Conference (EuRAD), 2012 9th European*, 2012, pp. 150–153 [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6450696. [Accessed: 28-Jan-2014]

[85] F. Glover, ‘Tabu Search—Part I’, *ORSA J. Comput.*, vol. 1, no. 3, pp. 190–206, Aug. 1989 [Online]. Available: http://pubsonline.informs.org/doi/abs/10.1287/ijoc.1.3.190. [Accessed: 28-Apr-2015]

[86] F. Glover, ‘Tabu Search—Part II’, *ORSA J. Comput.*, vol. 2, no. 1, pp. 4–32, Feb. 1990 [Online]. Available: http://pubsonline.informs.org/doi/abs/10.1287/ijoc.2.1.4. [Accessed: 28-Apr-2015]

[87] J. A. Högbom, ‘Aperture Synthesis with a Non-Regular Distribution of Interferometer Baselines’, *\aaps*, vol. 15, p. 417, Jun. 1974.

[88] U. J. Schwarz, ‘Mathematical-statistical Description of the Iterative Beam Removing Technique (Method CLEAN)’, *\aap*, vol. 65, p. 345, Apr. 1978.

[89] U. J. Schwarz, ‘The method “CLEAN”—Use, misuse and variations’, in *Image formation from coherence functions in astronomy*, Springer, 1979, pp. 261–275 [Online]. Available: http://link.springer.com/chapter/10.1007/978-94-009-9449-2_30. [Accessed: 09-May-2016]

[90] G. Hislop, N. Sakar, and C. Craeye, ‘Direction Finding With MUSIC and CLEAN’, *Antennas Propag. IEEE Trans. On*, vol. 61, no. 7, pp. 3839–3849, Jul. 2013.

[91] S. F. Gull and G. J. Daniell, ‘Image reconstruction from incomplete and noisy data’, *Nature*, vol. 272, no. 5655, pp. 686–690, Apr. 1978 [Online]. Available: http://dx.doi.org/10.1038/272686a0

[92] J. Tsao and B. D. Steinberg, ‘Reduction of sidelobe and speckle artifacts in microwave imaging: the CLEAN technique’, *Antennas Propag. IEEE Trans. On*, vol. 36, no. 4, pp. 543–556, 1988 [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1144. [Accessed: 09-May-2016]

[93] R. O. Schmidt, ‘Multiple emitter location and signal parameter estimation’, *Antennas Propag. IEEE Trans. On*, vol. 34, no. 3, pp. 276–280, 1986 [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1143830. [Accessed: 04-Dec-2014]

[94] Knowles Electronics, 'Digital Zero-Height SiSonicTM Microphone With Multi-Mode And Ultrasonic Support'. Knowles Electronics LLC, 16052014 [Online]. Available: http://www.knowles.com/eng/Products/Microphones/SiSonic%E2%84%A2-surface-mount-MEMS. [Accessed: 05-Sep-2017]

APPENDIX A

Initial MIMO Experimental System Development

A set of experiments has been conducted in order to prove the MIMO technique in radar context applicability with the MISL's existing equipment. A secondary goal of these experiments was getting around with the equipment and actually coming up with a configuration that can be used as a MIMO demonstrator test bed.

A sequential set of experiments has been planned starting with a single transmit single receive simple radar case and adding up elements until building up the fully functional MIMO case. Note that this report requires the knowledge of conventional phased array radars, digital signal processing, digital beamforming, and basic multiple-input multiple-output techniques used in radar context.

Plan

A plan of experiments including dates and goals have been prepared before experiments started. An executive summary of the plan is below in Table 0-1.

Table 0-1 Summary table of experiment plan

#	Task	Start	Deadline
1	1 Transmitter, 1 Receiver, 1 Target	25-Apr-14	05-May-14
2	2 Transmitters, 1 Receiver, 1 Target	05-May-14	12-May-14
3	1 Transmitter, 4 Receivers, 1 Target	12-May-14	19-May-14
4	1 Transmitter, 4 Receivers, 1 Target, Lambda/2 Spacing	19-May-14	09-Jun-14
5	1 Transmitter, 4 Receivers, 2 Targets, Lambda/2 Spacing	09-Jun-14	16-Jun-14
6	2 Transmitter, 4 Receivers, 1 Targets, Lambda/2 Spacing	16-Jun-14	07-Jul-14
7	2 Transmitter, 4 Receivers, 2 Targets, Lambda/2 Spacing	07-Jul-14	28-Jul-14

The motivation for task 1 is simply being able to detect a target's existence and measure its range. Motivation for task 2 is to observe and verify the expected phase difference between received signals. For task 3 it is almost the same thing but it is to test it via moving the receiver. Task 4 is about emulating a phased array with appropriate element spacing and verifying the process of receive beamforming. Task 5 is similar to task 4 but it includes testing the ability to resolve two targets in spatial domain (range and cross-range). Task 6 is finally building a MIMO and applying both transmit and receive beamforming and testing the ability to detect a target. Finally, task 7 is similar to task 6 but it includes testing the ability to resolve two targets in spatial domain and compare it with results of task 5.

Setup

Equipment

A list of all the equipment used can be found below with their outstanding properties.

- Anechoic Test Chamber
    - Sufficiently RF absorbent at least up to 4 Ghz
- Tektronix AWG7102 (Arbitrary Waveform Generator)
    - 2 Channel Output
    - External Trigger support
    - Adjustable sample frequency
    - 10 GigaSamples/sec/channel (two channels)
    - 20 GigaSamples/sec (one channel only)
- Tektronix DPO72004C (Digital Phosphorus Oscilloscope)
    - 4 Channel Input
    - External Trigger support
    - Continuous Fast Frame capture with respect to trigger events
    - Adjustable sample frequency
    - 50 GigaSamples/sec/channel
    - 16 GHz hardware low-pass filter
- Omnidirectional 2-Pence Antenna
    - Bandwidth: N/A, assumed wideband around 3 Ghz
    - Beamwidth: N/A, Assumed omnidirectional on azimuth
    - Gain: ~3 dBi
- Ultra Wide Band Horn Antenna (white)
    - Bandwidth: 1-18Ghz
    - Beamwidth: 60 degrees around 3 Ghz
    - Gain: ~8 dBi
- USB Pulse Generator
    - Up to 100Mhz pulse generator
    - Adjustable duty cycle
- Microwave cables
- Minicircuits mixers, filters etc.

APPENDIX A

Connections

In order to make the equipment work as an active coherent radar we had to do some adjustments to our systems. The problems we faced and their solutions can be summarized as below.

DPO and AWG synchronous triggering and frame capturing is problematic due to several reasons. The first reason is that trigger source is not precise to the nanosecond level. Secondly, trigger processing of AWG and DPO are not consistent with themselves, meaning that they introduce pseudorandom delays to frames. DPO also introduces different delays to different channels making it a lot more difficult to try and find a fixed pattern and match it. Our solution to this problem was to use a reference point in the received signals and adjust the timestamps of the signals with respect to these reference points (i.e. post-acquisition software processing). We took advantage of having our transmitter and receiver being separate. So our reference point became the direct path transmission of the radar signal. It has very good SNR and in a practical application this is the one path that we know that is not going to change. For non-simultaneous transmit/receive radars this technique can still be used by using a high-reflectivity close-range known target.

Due to calibration issues, DPO introduces destructive and non-filterable interference when it's used with sample frequencies higher than 3.125 Gsps. After a series of tests and a firmware upgrade, we've concluded that it is probably best to use it with 3.125 Gsps configuration until it's properly calibrated. This has introduced some others problems that we were able to solve easier. We needed to downconvert our radar waveform which was around 3Ghz to a frequency that we can still capture with low bandwidth. We have used mixers and a low-pass filter to downmix the received signal and we brought the received signal to under 1.6 Ghz bandwidth limit to meet sampling criteria. As the local oscillator signal we had to use the second channel of AWG since we were not able to use an external oscillator coherently with AWG.

Due to lack of parts and in the name of consistency, we decided to go with a time orthogonal MIMO, which allowed us to use the same transmitter and receiver as many times as we like within different time intervals. Basically, after each measurement we have moved the receiver or transmitter antenna as needed. A final simplified diagram of the setup is demonstrated in Figure 0-1.

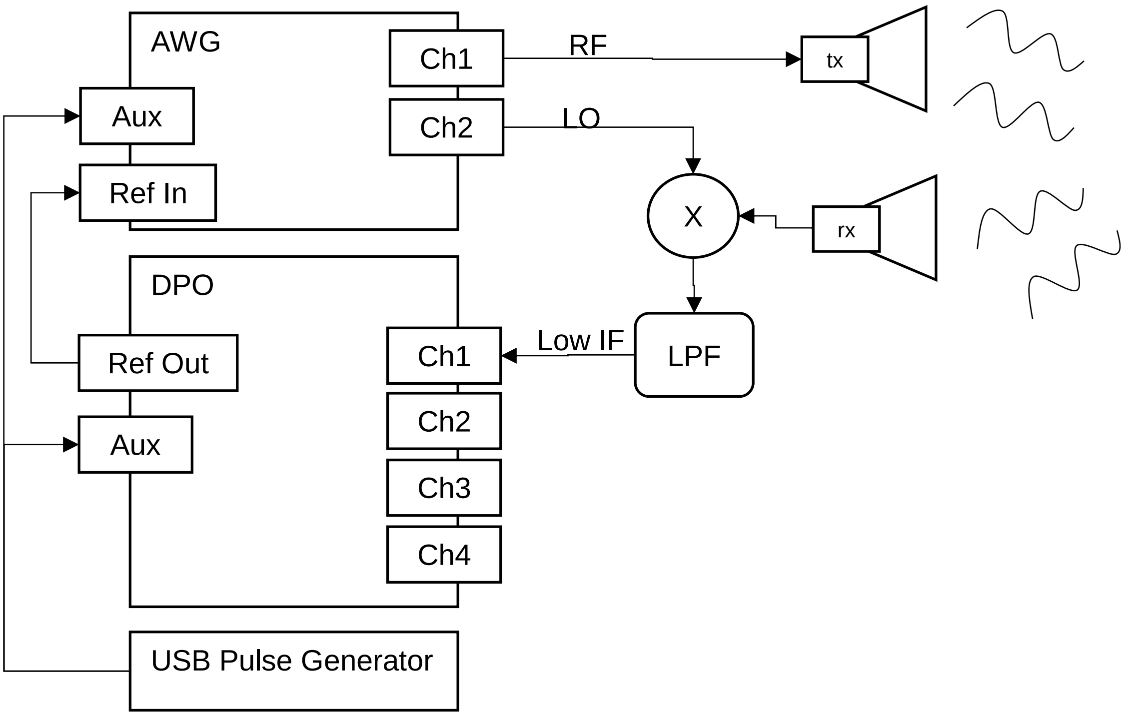


Figure 0-1 Demonstrator system block diagram

The received signals at low intermediate frequency were captured with the fast frame property of the DPO, so when recording starts all transmitted signals' reflections were recorded without gaps. Since the initial experiments only included stationary targets, this is not of very importance. However it was essential to speed up the acquisition process and increase the signal-to-noise ratio without too much effort.

Waveform

In most of the experiments an up-chirp LFM waveform has been used to take advantage of range compression. Mostly this waveform had a 900 Mhz bandwidth with a rectangular envelope with zero sidelobe reduction but with 15 cm theoretical range resolution. Other types of waveforms such as Gaussian enveloped single frequency pulses and Gaussian enveloped chirps has also been tried but retired due to inability to acquire appropriate SNR and/or range resolution.

Software Pre-processing

As mentioned above a pre-processing has been done in software to correct the timings of the received signals. Short explanation of the all preprocessing steps is, cutting off DC, band-pass filtering, matched filtering with the reference signal and determining the peak locations. And finally all data frames have been adjusted to start from their own first peak location (initial data is discarded). This pre-processing gives good enough results to go on with further processing. The difference between misaligned data and pre-processed data in means of range-Doppler profile and in slow time domain can be observed in Figure 0-3 and Figure 0-5.

Multiple Transmitter/Receiver Configurations

The equipment at hand currently allows us to operate only a single transmitter and a single receiver simultaneously. But by simply moving the transmitter and/or receiver around the test area with precise distances and taking the measurements again we can emulate the equivalent hardware in time domain. While this method allows us to emulate test beds that might not have been possible with resources at hand, drawback of this method is not being able to do precise Doppler measurements. Since we have specialized MIMO equipment coming in and Doppler measurements are not priority right now, this drawback was not an issue.

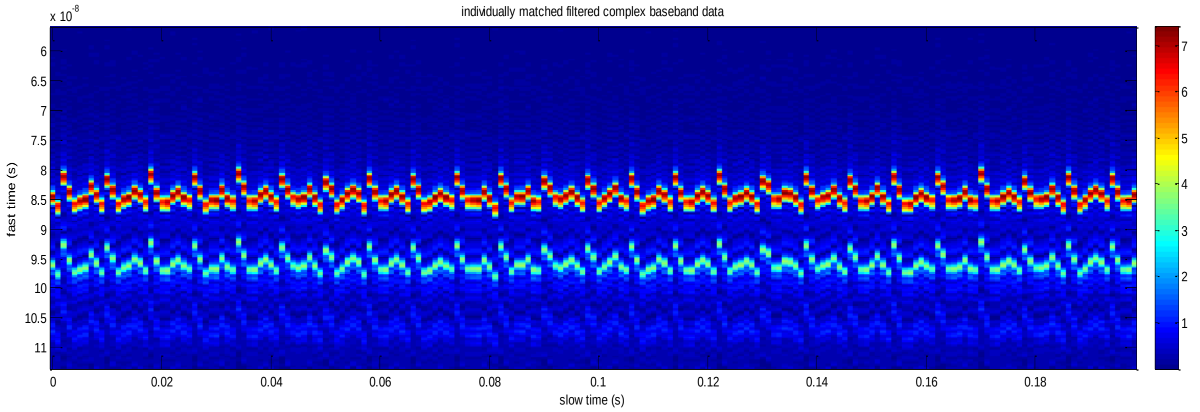


Figure 0-2 Slow time plot of uncorrected data after matched filtering (single stationary target)

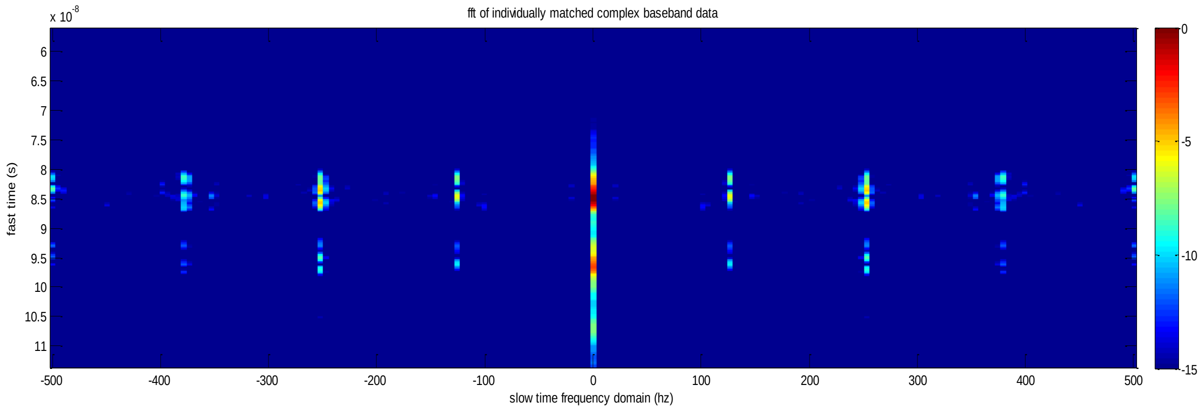


Figure 0-3 FFT of uncorrected data after matched filtering (single stationary target)

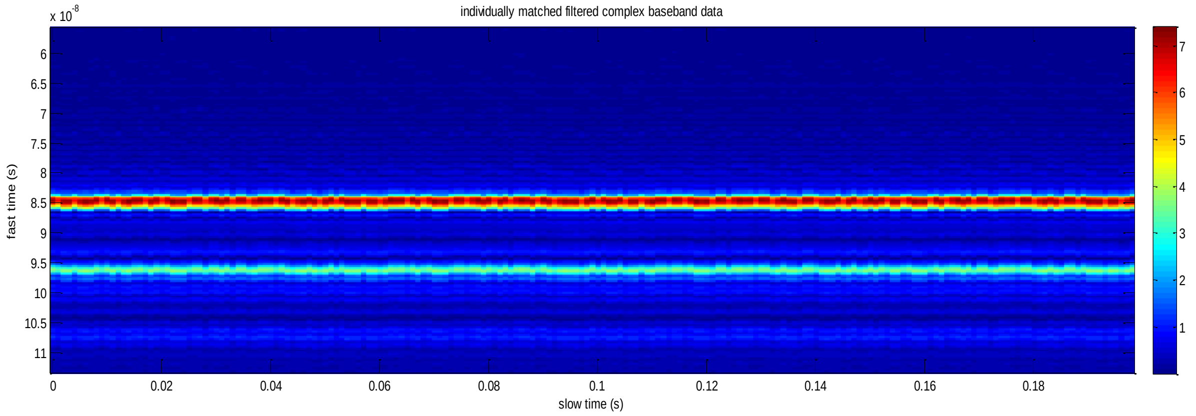


Figure 0-4 Slow time plot of pre-processed data after matched filtering (single stationary target)

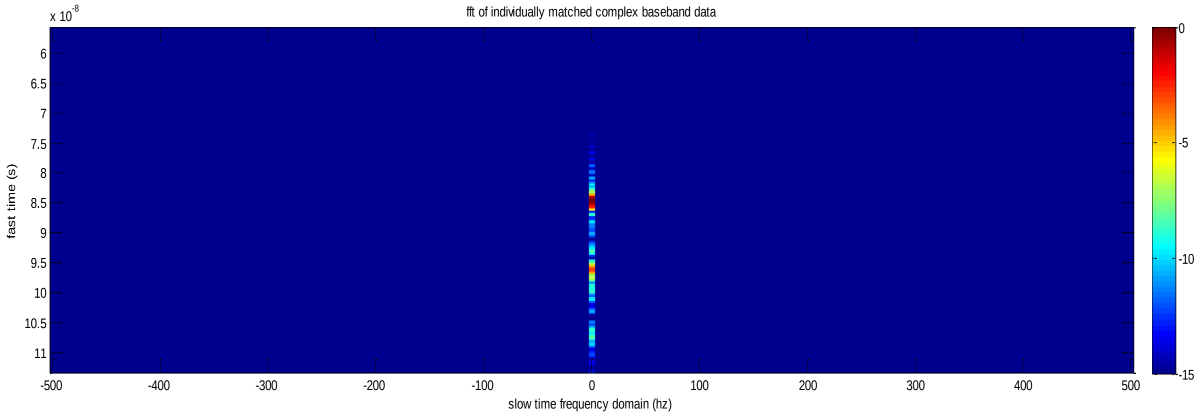


Figure 0-5 FFT of pre-processed data after matched filtering (single stationary target)

Results

In this section results will be explained briefly with the scenario descriptions backed up with photos and resulting plots. The numbers in headings M,N and K are references to element numbers greater than 1, and Lambda refers to the wavelength of the waveform that is used. During the experiments some shorthand names were used to define target types and locations of targets during experiments. These locations are explained as the table below. Also when directions are mentioned it should be taken account that "North" is not actual Earth North but it is the direction from the first pole to the second pole of the anechoic chamber. An approximate map explaining positions and directions with the last definition can be found in

Table 0-3.

Table 0-2 Shorthands used in experiments and their descriptions (the distances are approximate).

Shorthand	Description
Middle	A Point in the middle of the anechoic chamber, at 2 m north to pole seen in Figure 0-6.
Pole	Second pole in the anechoic chamber, usually at 3 m north to first pole. This is NOT the pole where our transmitters/receivers are next to.
Door	The second door in the anechoic chamber, at 3 m north, 1.5 m west to first pole.

Left	A close point in the same vertical grid as the Door location. 2 m north, 1.5m west to first pole.
Sheep	Aluminium covered 100x30 cm rectangular target
Shield	Metal 150x40 cm rectangular target

Table 0-3 A table demonstrating the anechoic chamber as a grid, each cell represents a 50x50 cm^2 square in the test chamber

2nd Door			2nd Pole	
Left			Middle	
	↑ North			
	↓South			
	←West	→East		
1st Door	TX		1st Pole	

Couple of photos from several of the configurations can be observed below in Figure 0-6, Figure 0-7, Figure 0-8, and Figure 0-9.

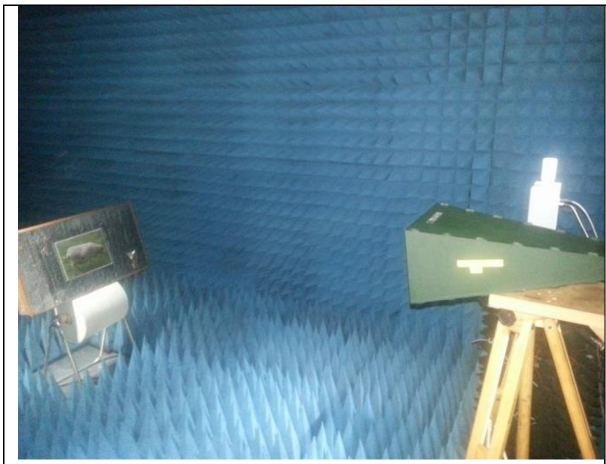


Figure 0-6 SheepMiddle: a photo from the first setup (target is ~2 m away).

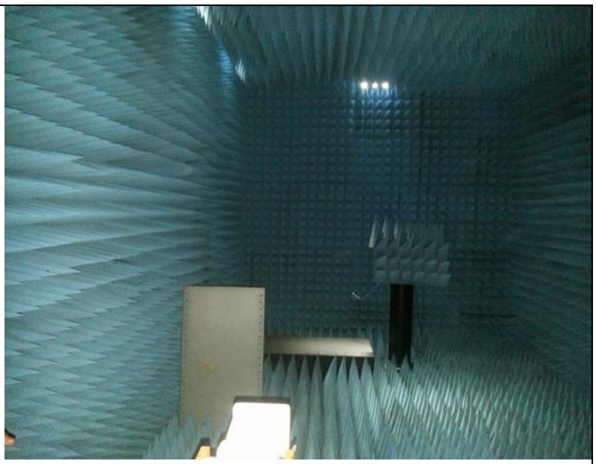


Figure 0-7 ShieldLeft: pole is covered like in the photo unless specifically specified

Figure 0-8 ShieldDoor+SheepPole: pole is always uncovered when a target on pole is present

Figure 0-9 SheepMiddle+BallPole

1-Transmitter, 1-Receiver, 1-Target

A directional transmitter and an omnidirectional receiver have been set up in the anechoic chamber with close enough proximity to assume transmitter and receiver are at single point. And target(s) are placed within the anechoic chamber with known ranges to the radar. Below in the Figure 0-10 are the results of various configurations of single-transmitter single-receiver experiment. One of the first things to observe is the decrease in returned power as range increases. We were simply able to detect targets at specified ranges.

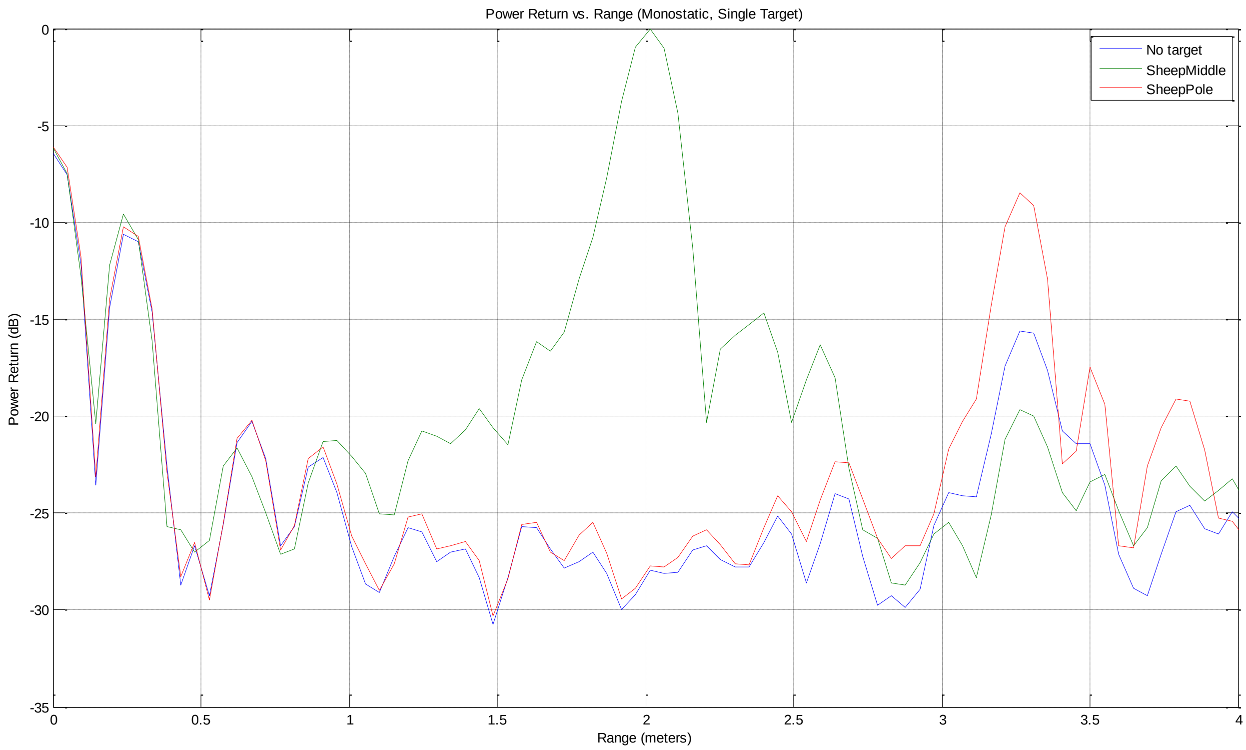


Figure 0-10 Range measurements from task 1

N-Transmitters, 1-Receiver, 1-Target

The setup started similar to first setup. After a set of measurements, transmitter has been moved about 20 cm to west of its original position. After another set of measurements has been done, the transmitter has been moved another 20 cm to west of its position, resulting in emulation of 3 transmitters. The results are merged via software pre-processing mentioned earlier. Below are the results with a single target when transmitter separation is about 20 cm. Beamforming was not tried since grating lobes were expected.

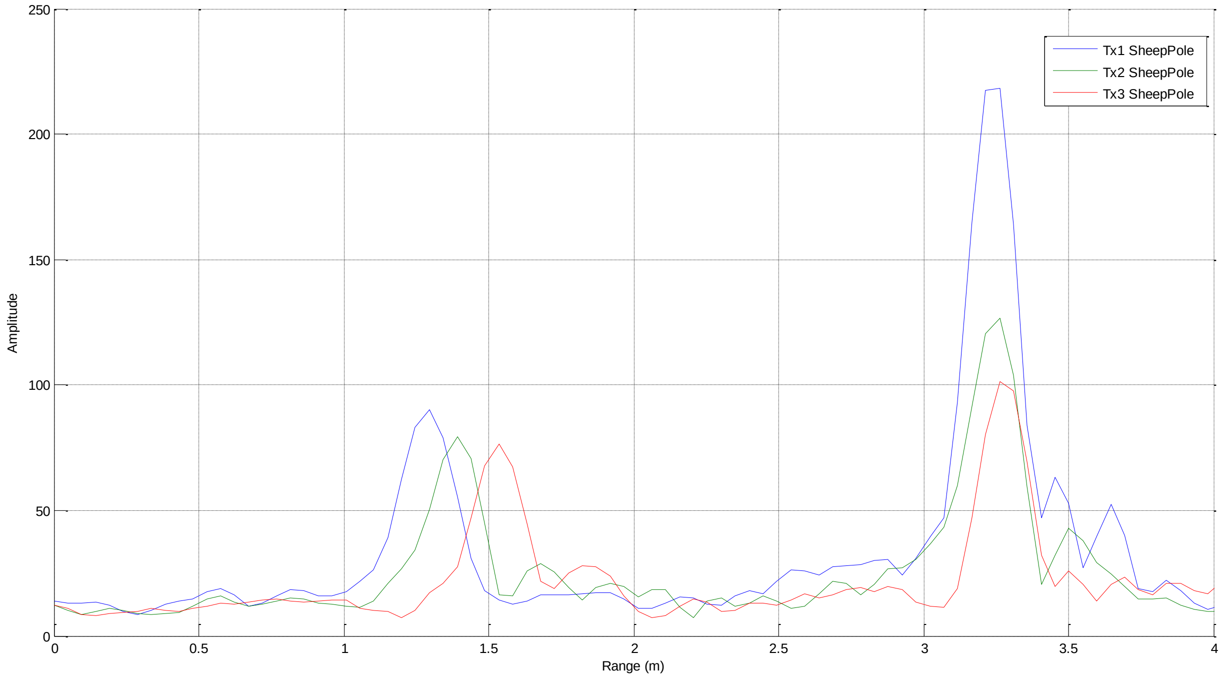
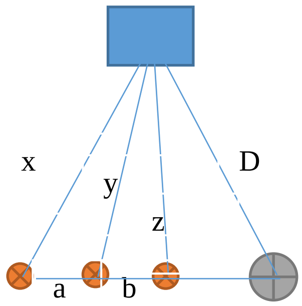


Figure 0-11 Range measurements from task 2, first set of peaks are the results of direct path transmission

With this experiment we were able to observe expected phase differences and time delays with a precision higher than hoped (but was expected). With processing the distance between the first peaks (direct path) was set to the distance which was expected. And naturally the second peaks (target returns) aligned on each other. Geometrically the expected outcome can be explained as in the figure below.



Time and phase difference between direct path transmissions should be directly proportional to *b* and *b*+*a*. However time and phase difference between target reflections should be directly proportional to *D*+*z*, *D*+*y*

Figure 0-12 Geometrical explanation of the expected and obtained results.

1-Transmitter, N-Receivers, 1-Target

The setup is very similar to previous task, instead of moving the transmitter this time we have moved the receiver. Two receiver locations have been used and receiver separation in this experiment is about 56 cm (manually measured). Below are attempts to beamform in Figure 0-14 and Figure 0-16. As expected we observe grating lobes, but we still observe peaks at expected points as well. In this experiment -different from others- a Gaussian enveloped LFM chirp has been used with 300 Mhz bandwidth allowing a theoretical range resolution of ~50 cm. When target is at pole, it has been calculated that the target stands at ~22 degrees, and when target is at the middle it has been calculated that it stands at ~33 degrees.

The computed azimuth pattern of an array with 2 elements and with 56 cm (corresponding to a 5.6 wavelength for 3.1 GHz carrier) is shown below in Figure 0-13. Even though the transmitter separation was almost random and much greater than half wavelength, it was a good test to test beamforming process. We were able to observe the expected range and angular patterns in our measurements (See Figure 0-13, Figure 0-15 and Figure 0-17).

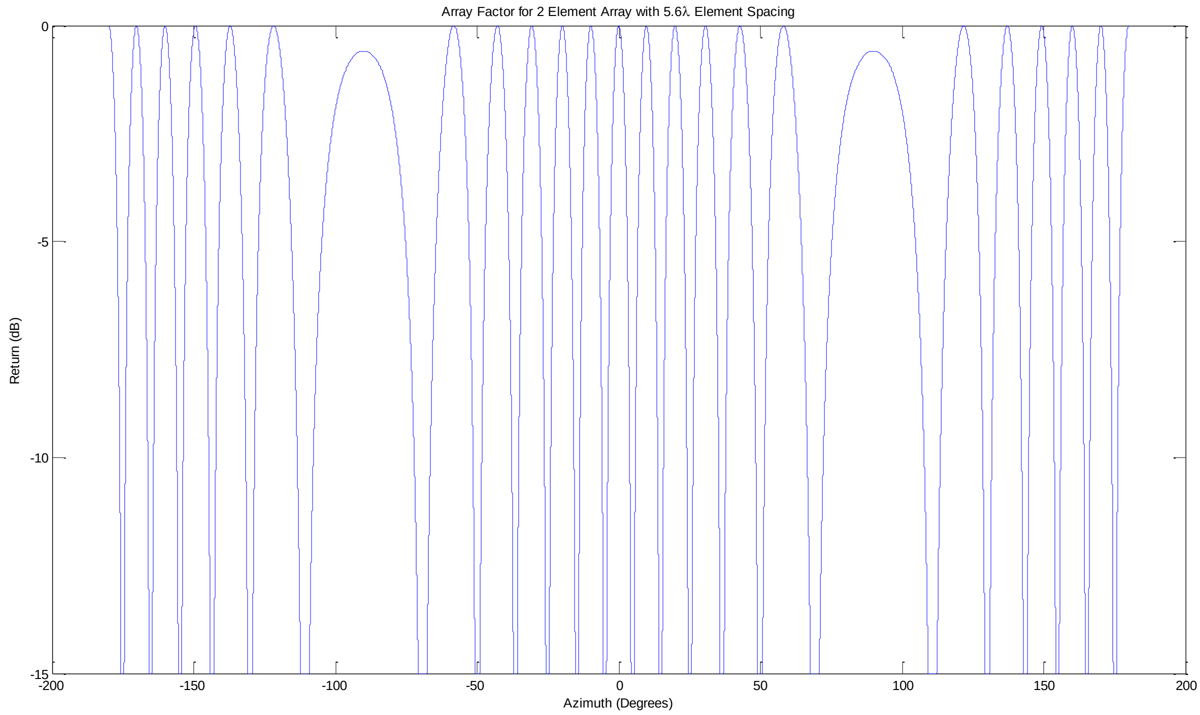


Figure 0-13 Array Factor of phased array with 2 elements and 5.6 lambda element spacing

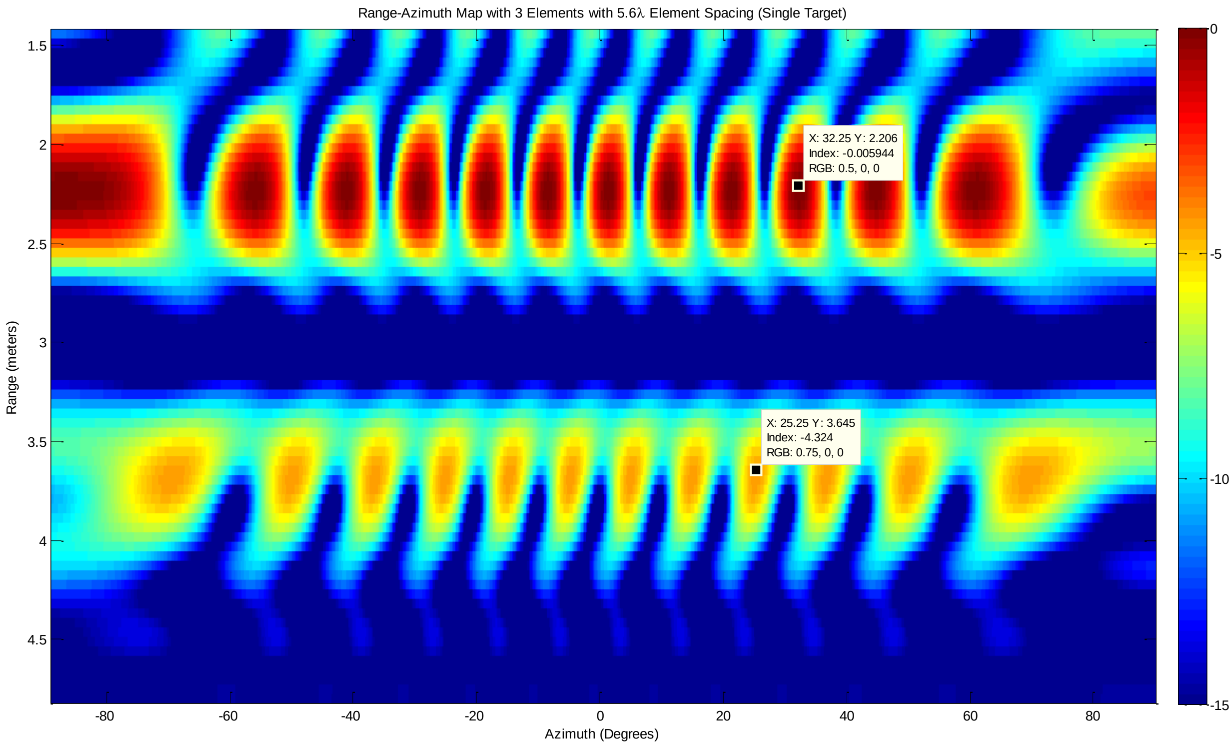


Figure 0-14 Range vs. Azimuth plot of the test chamber from task 4, SheepMiddle; the second target at 3.75 meters is assumed to be a reflection from the 2^{nd} pole in the test chamber

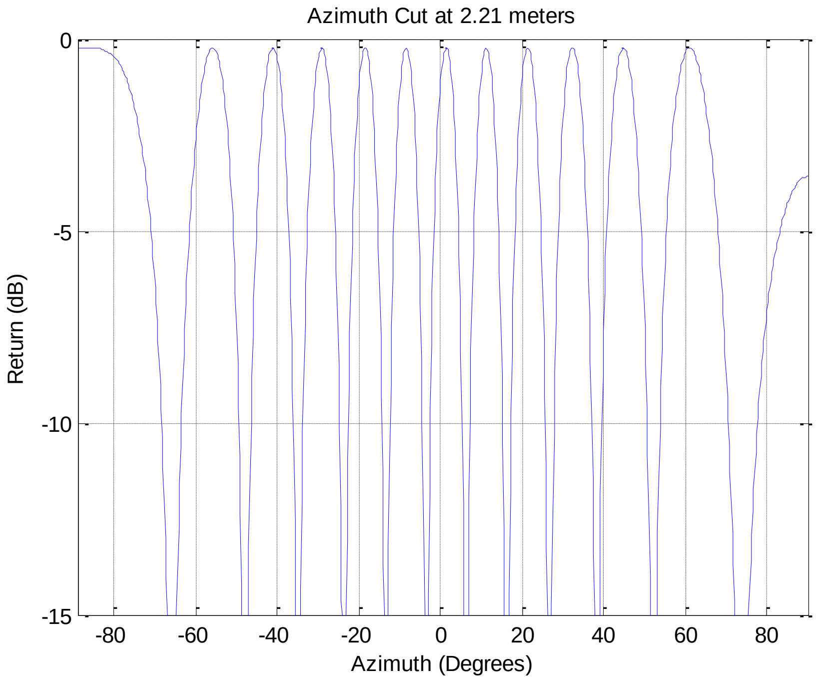


Figure 0-15 Azimuth cut of measurement at 2.2 meters when target is at the middle

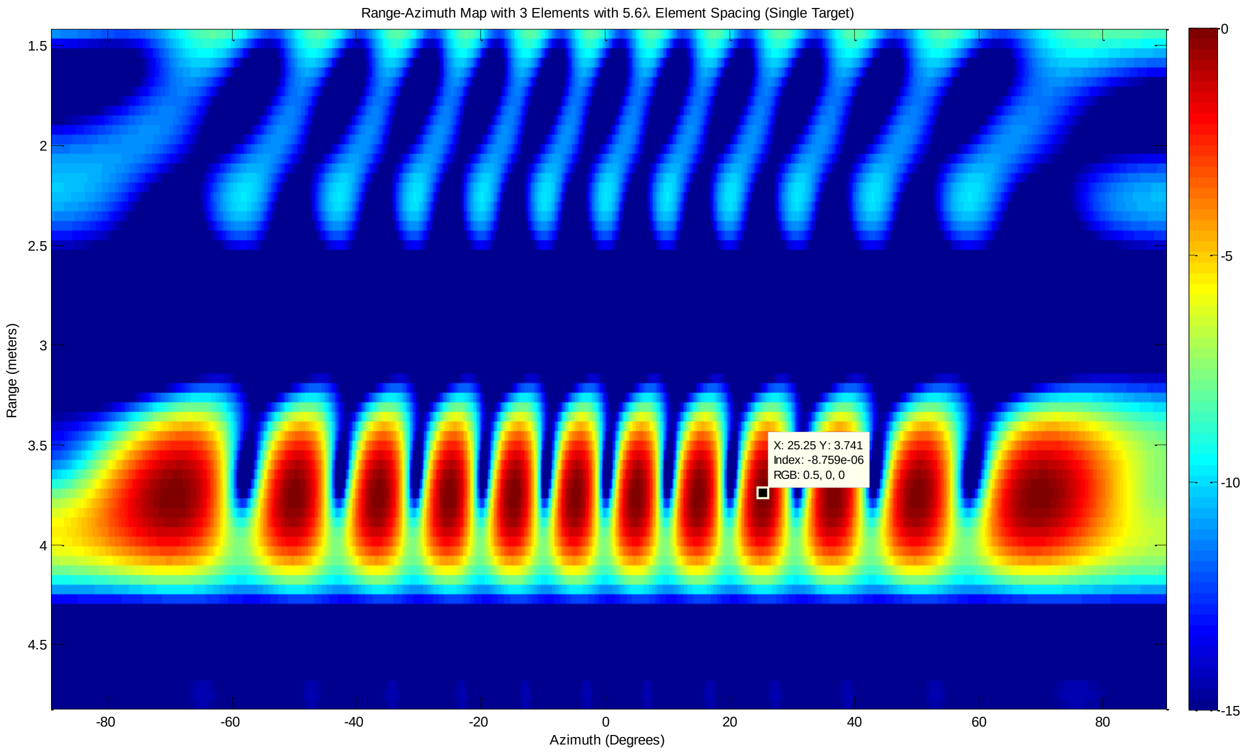


Figure 0-16 Azimuth vs. Range plot (map) of the test chamber from task 4, SheepPole;

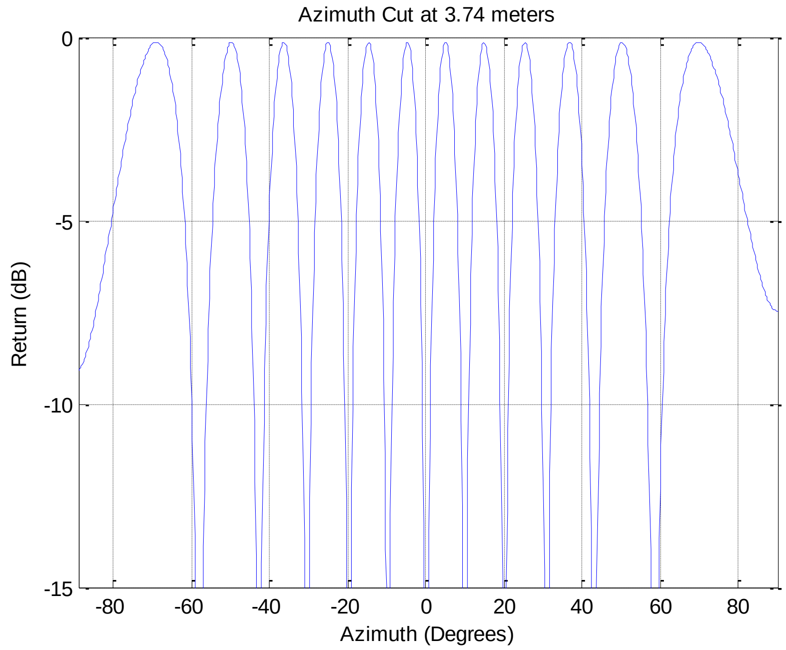


Figure 0-17 Azimuth cut of measurement at 3.7 meters when target is at pole

1-Transmitter, N-Receivers, 1-Target, Lambda/2 Spacing

This experiment was performed very similar to previous one, only this time the receiver separation was set to a mere 4 cm which just a little less than our waveform's centre frequency half wavelength (3.5 Ghz – 4.3 cm). This allowed us to beamform without grating lobes, resulting in a single peak per target. The waveform used is a 1 GHz bandwidth rectangular LFM up-chirp with 3.5 GHz carrier. Expected range resolution is ~15 cm. For comparison the expected array pattern for a 9-element phased array is shown in Figure 0-18. The resulting range-azimuth map and its range and azimuth cuts at maximum point can be found in Figure 0-19 and Figure 0-20 and Figure 0-21 respectively. In this experiment we were actually able to resolve a single target as a single peak in our map even with relatively high angular sidelobes.

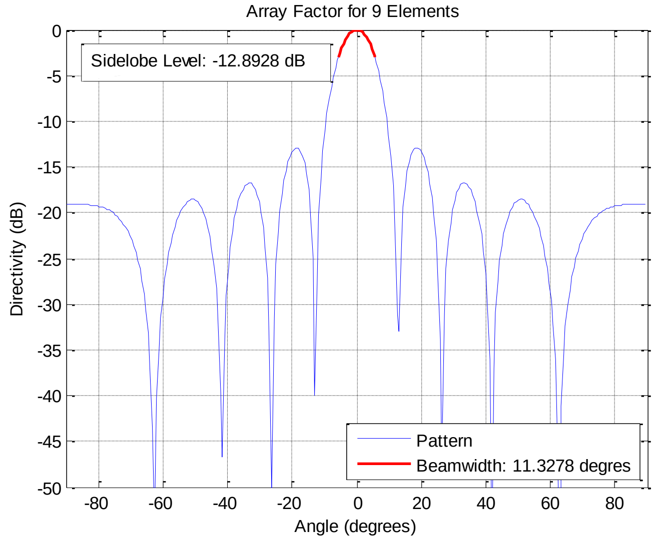


Figure 0-18 Pattern for phased array with 9 elements with half wavelength spacing

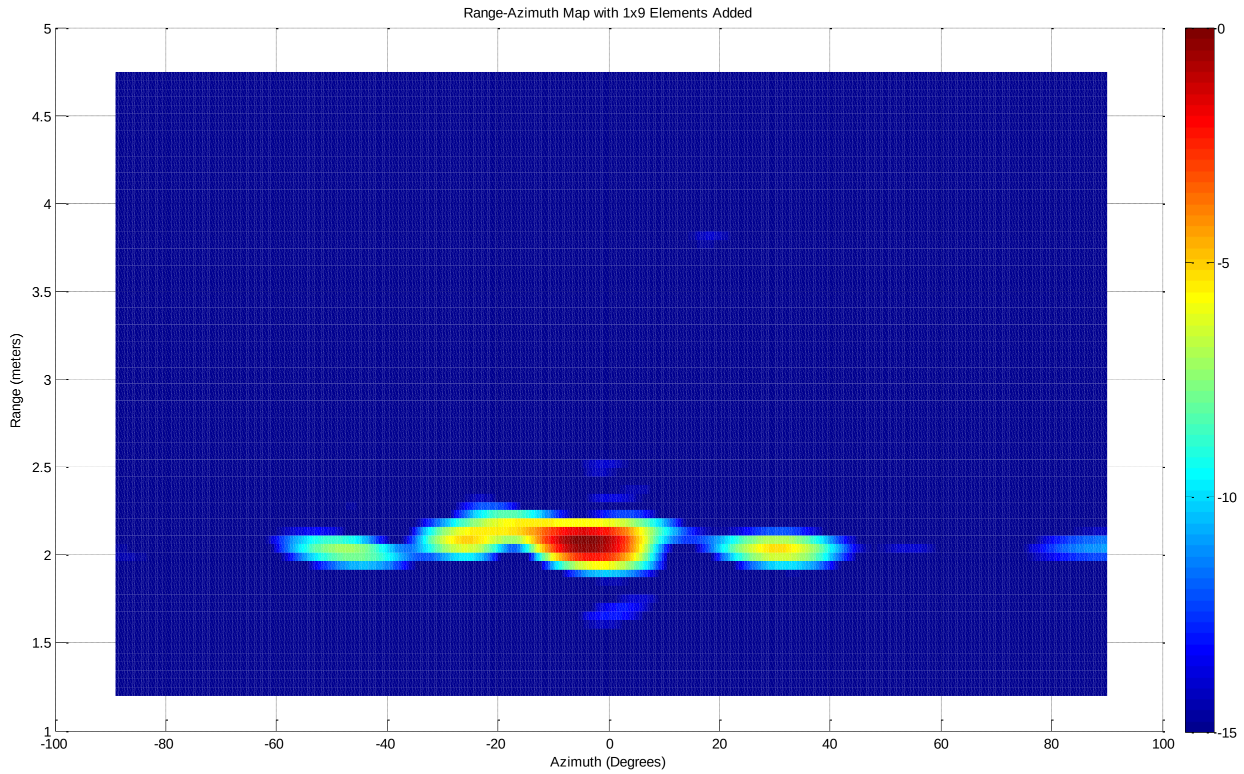


Figure 0-19 Range-Azimuth map of the test chamber from task 5 with single target, ShieldLeft

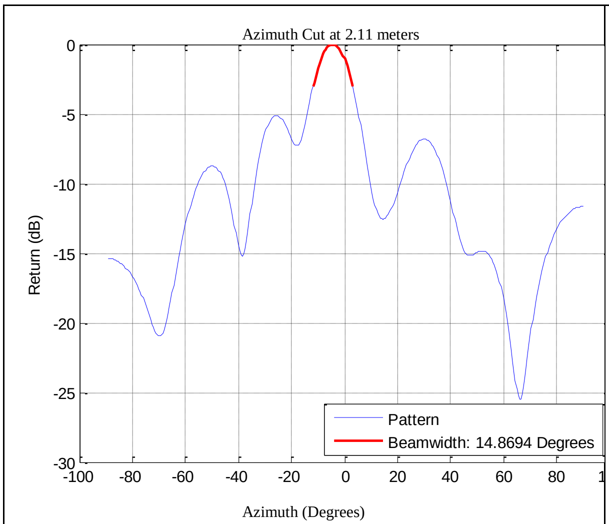


Figure 0-20 Azimuth cut of map with single target

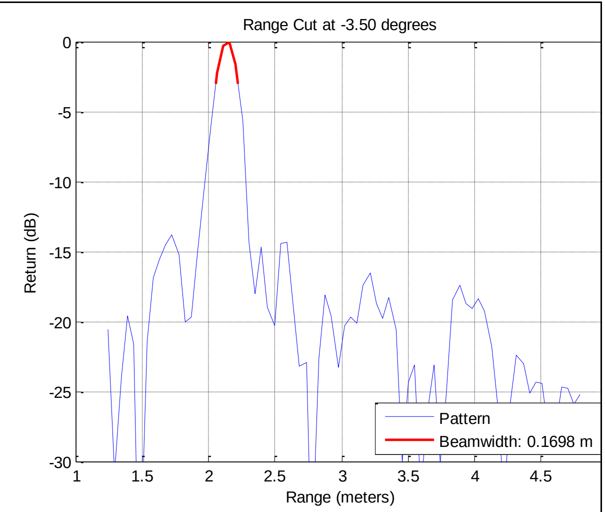


Figure 0-21 Range cut of map with single target

1-Transmitter, N-Receivers, K-Targets, Lambda/2 Spacing

This experiment was performed exactly the same as previous experiment except more targets were used and tried to resolve in the map. In Figure 0-22 and Figure 0-25 you can observe multiple target resolution. And the azimuth and range cuts of these plots at the global maxima can be found in Figure 0-23, Figure 0-24, Figure 0-26 and Figure 0-27 respectively.

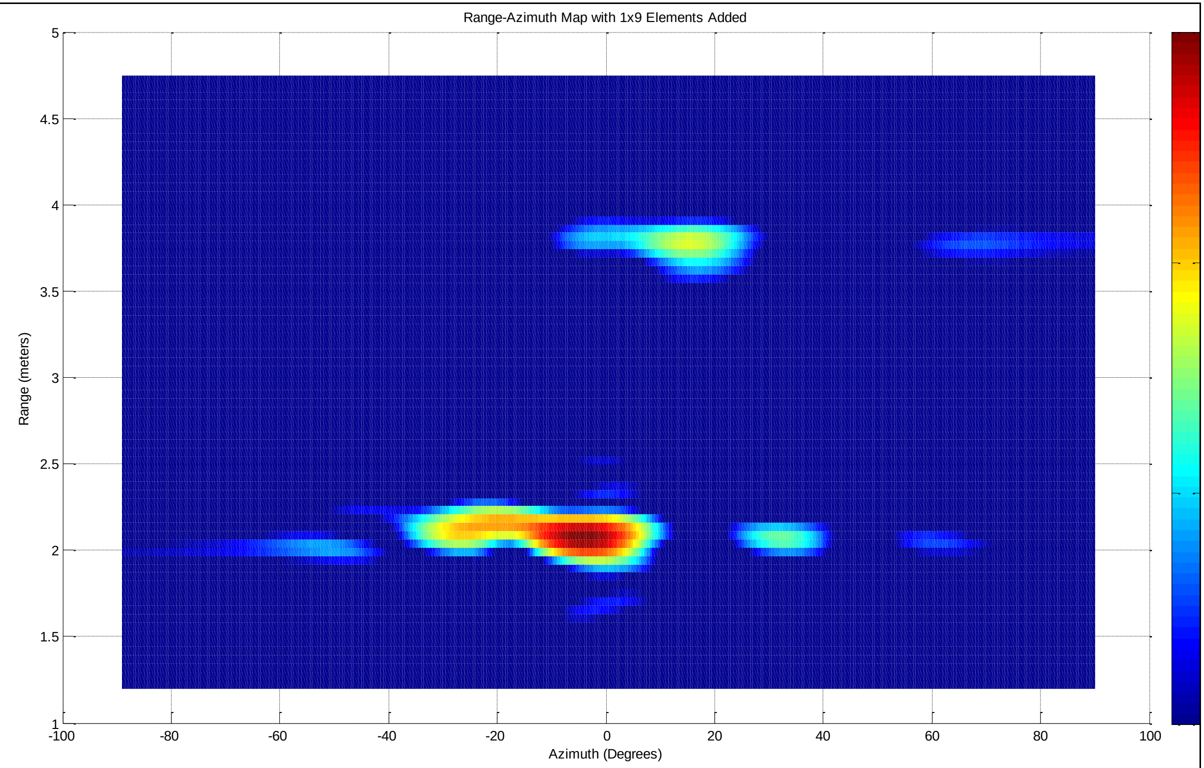


Figure 0-22 Multiple target resolution with multiple receive antennas, ShieldLeft+SheepPole

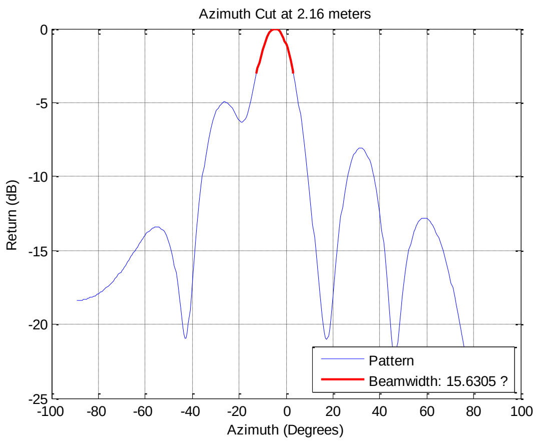


Figure 0-23 Azimuth cut of map with two targets

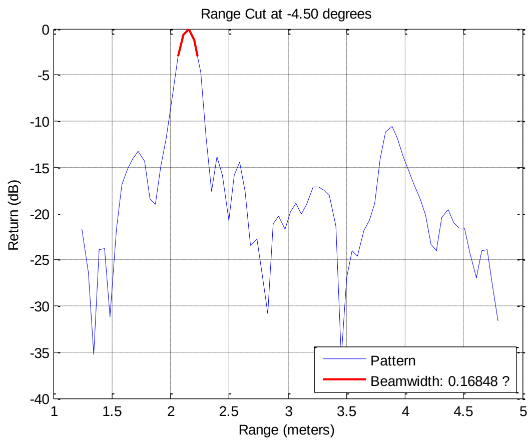


Figure 0-24 Range cut of map with two targets

With this much range difference between targets, we were able to resolve two separate targets. We assume that sidelobes have formed to the left of the targets due to improper spacing between elements.

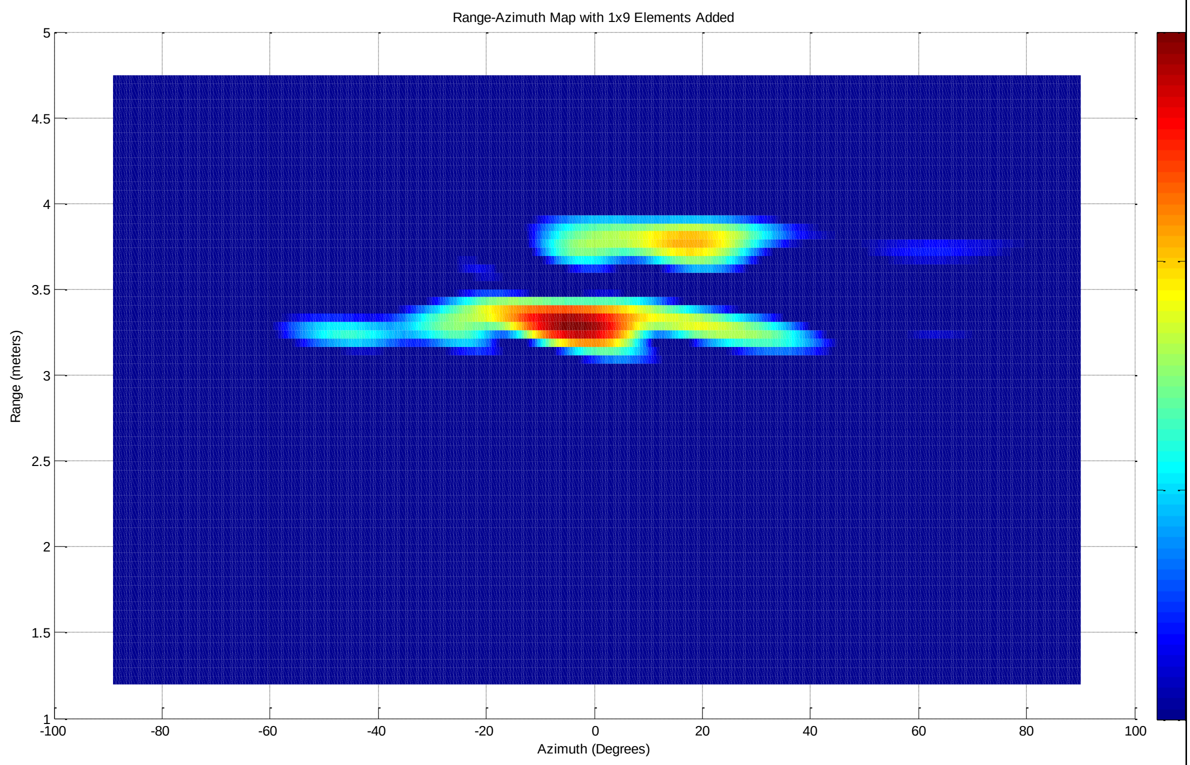


Figure 0-25 Multiple target resolution with multiple receive antennas, ShieldDoor+SheepPole

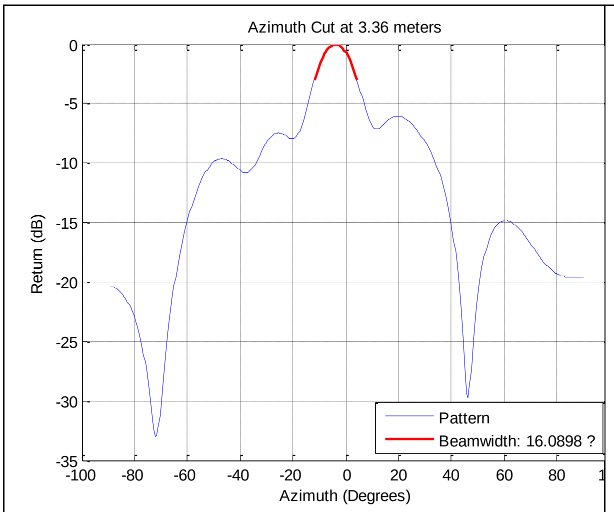


Figure 0-26 Azimuth cut of second experiment with multiple receiver, multiple targets

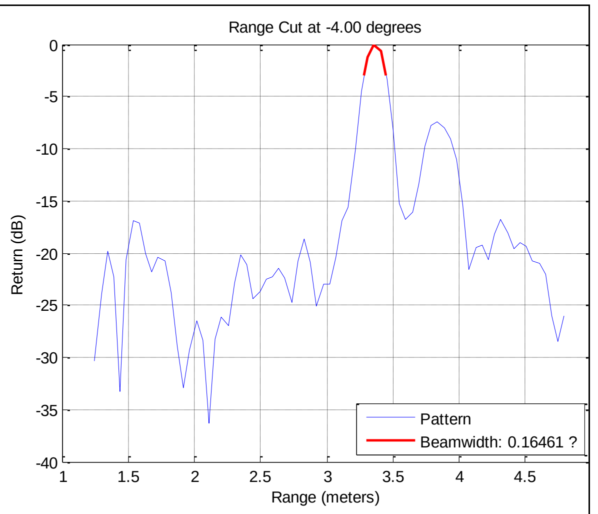


Figure 0-27 Range cut of second experiment with multiple receiver, multiple targets

This second setup was initially configured to test the ability to resolve two targets at the same range bin, but it seems we have failed to place our targets in the same range bin. This experiment's this part can be repeated either with more precise positioning of the targets or using a waveform with less range resolution.

M-Transmitter, N-Receivers, 1-Target, Lambda/2 Spacing

This experiment is basically the combination of multiple-transmitter and multiple-receiver experiments. And although this experiment required MxN measurements, this experiment is an emulation of the MIMO radar. For comparison the same configuration with task 4 is used to observe immediate effect of using 2 transmitters instead of 1. The angular resolution improvement is clearly visible in Figure 0-29 and we can say that it has halved as expected. Slices of the map at maximum point can be found in Figure 0-30 and Figure 0-31. For comparison theoretical radiation pattern of a 2x9 MIMO radar is given in Figure 0-28.

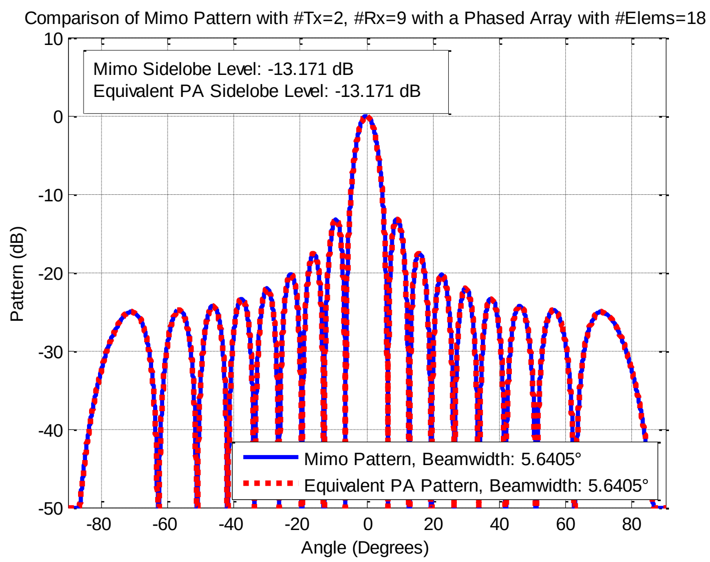


Figure 0-28 MIMO array pattern with 2x9 elements

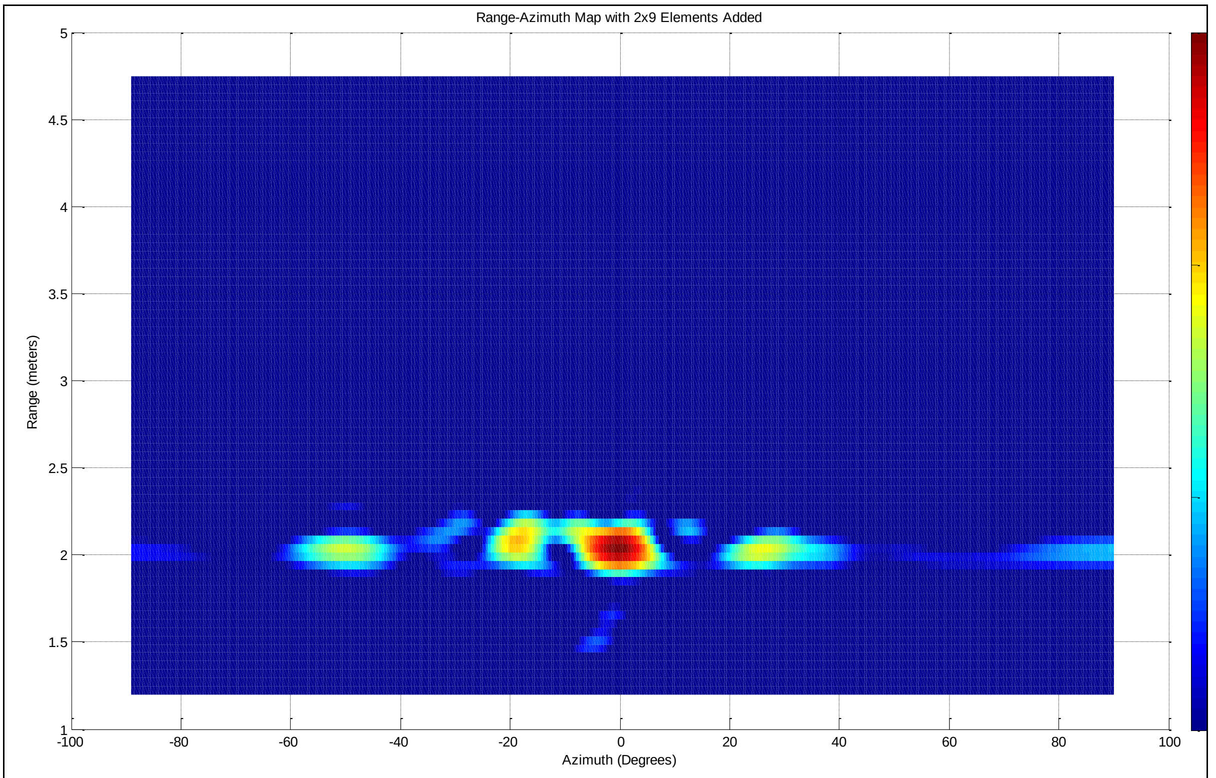


Figure 0-29 2-Transmitter x 9-Receiver MIMO Radar results, ShieldLeft

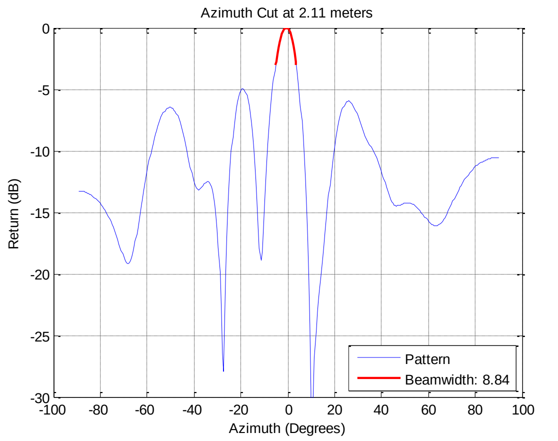


Figure 0-30 Azimuth cut of MIMO experiment with single target

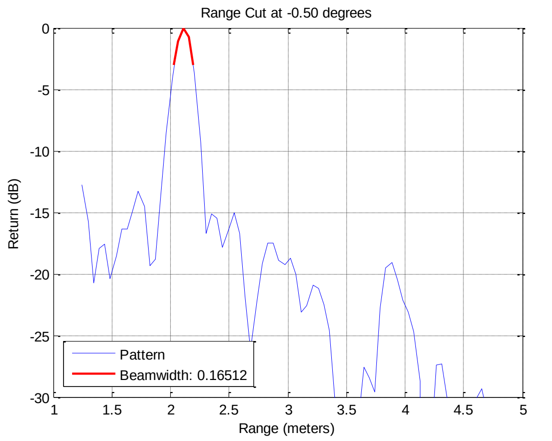


Figure 0-31 Range cut of MIMO experiment with single target

The increase in angular resolution is easily observed for our first MIMO experiment. Although the beamwidth is not as small as expected the improvement is exactly as expected. Also some sidelobes can be observed which are assumed to be because of improper spacing of elements.

M-Transmitter, N-Receivers, K-Targets, Lambda/2 Spacing

This is the same experiment as previous repeated with multiple targets instead of one. As before, the imminent observation is again the angular resolution. We assume that the left sidelobes are due to improper spacing between elements therefore damaging the MIMO beamforming process (i.e. receive null and transmit grating lobe mismatch). The range-azimuth maps of the test chamber configurations can be observed in Figure 0-32 and Figure 0-35. And the azimuth and range cuts of these plots at the global maxima can be found in Figure 0-33, Figure 0-34, Figure 0-36 and Figure 0-37 respectively.

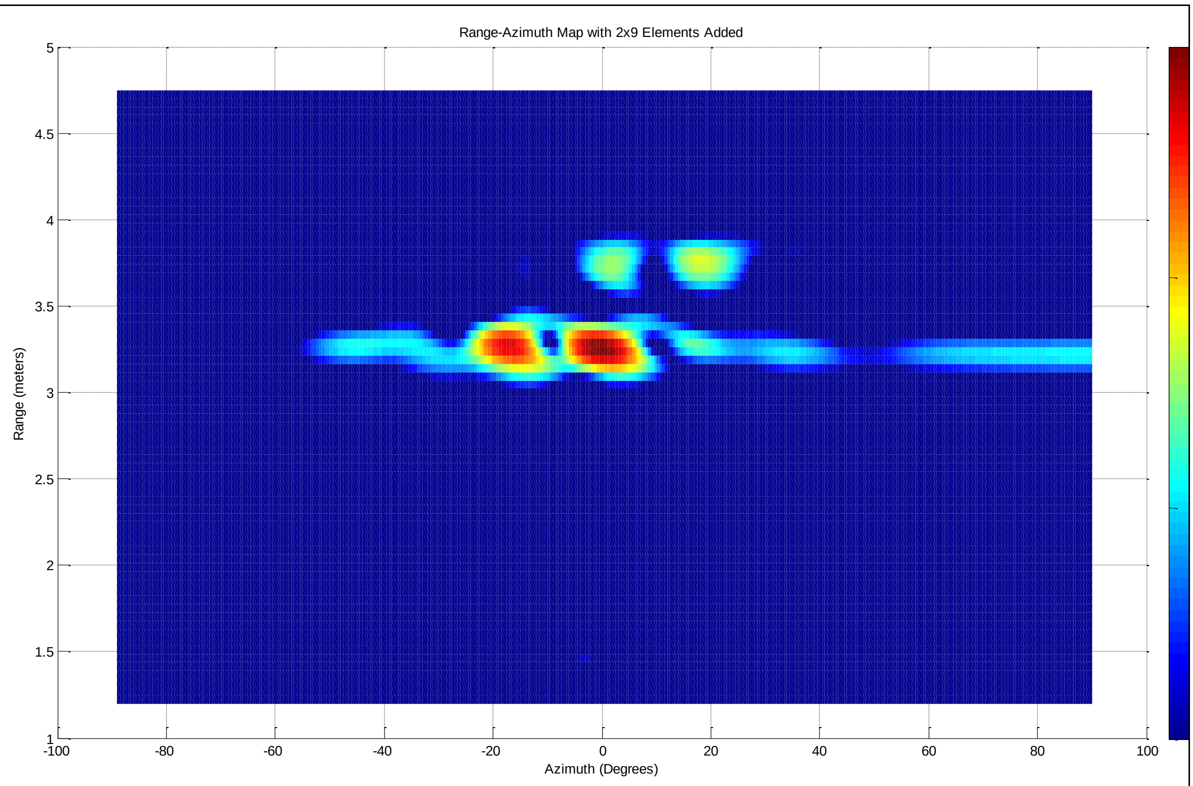


Figure 0-32 Emulated MIMO radar results with 2-transmitter and 9-receivers; ShieldDoor+SheepPole

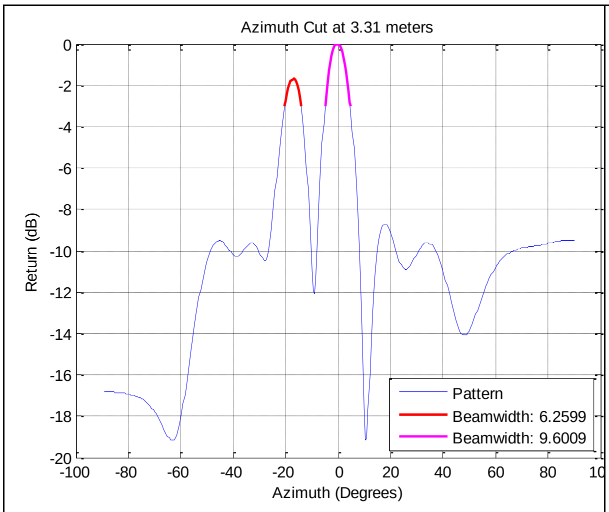


Figure 0-33 Azimuth cut of MIMO experiment with two targets

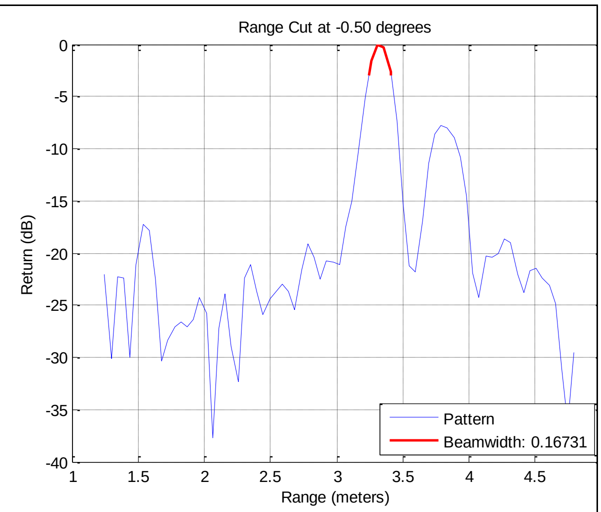


Figure 0-34 Range cut of MIMO experiment with two targets

Like its similar -single-transmitter, multiple receiver- we observe some sidelobes forming up, but we also observe the increase in angular resolution and better resolving of the targets in azimuth domain.

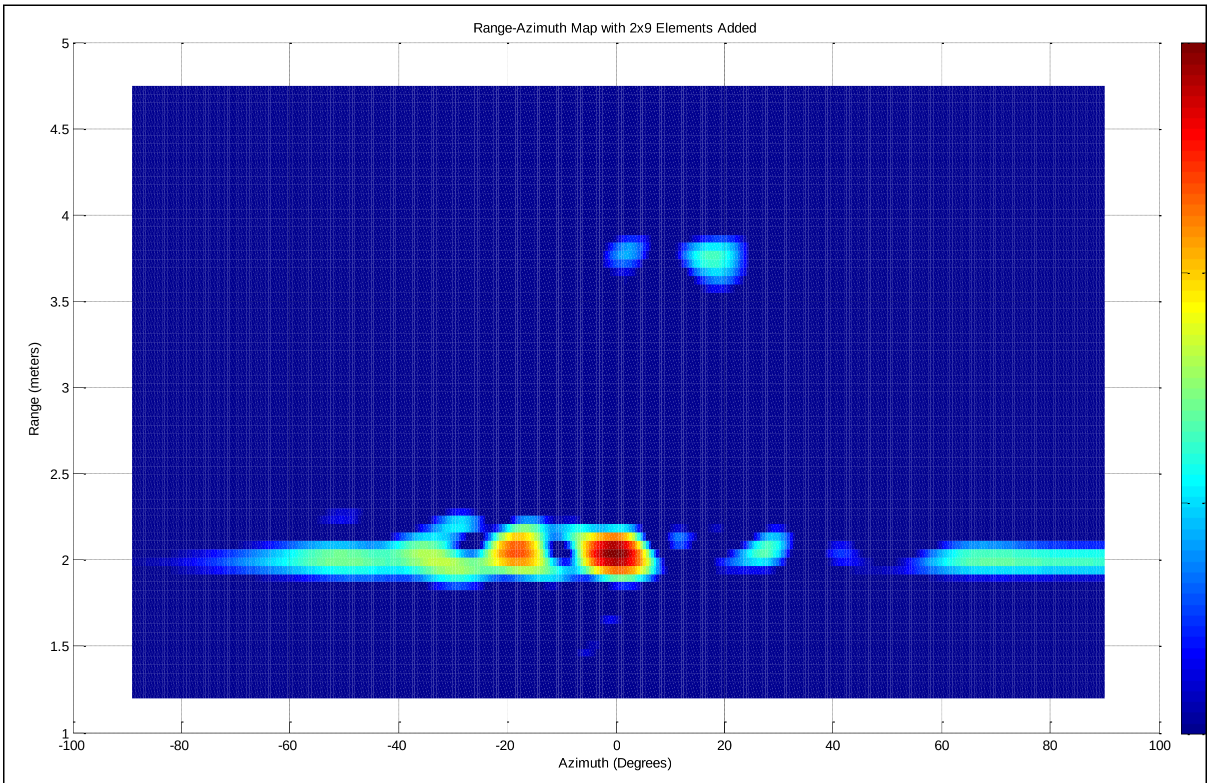


Figure 0-35 Emulated MIMO radar results with 2-transmitter and 9-receivers; ShieldLeft+SheepPole

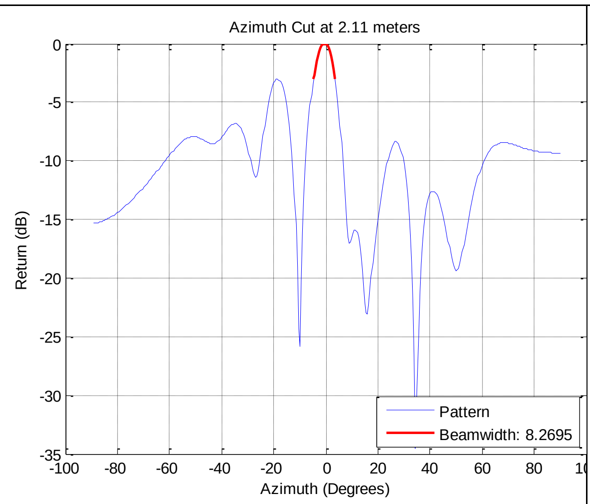


Figure 0-36 Azimuth cut of second MIMO experiment with two targets

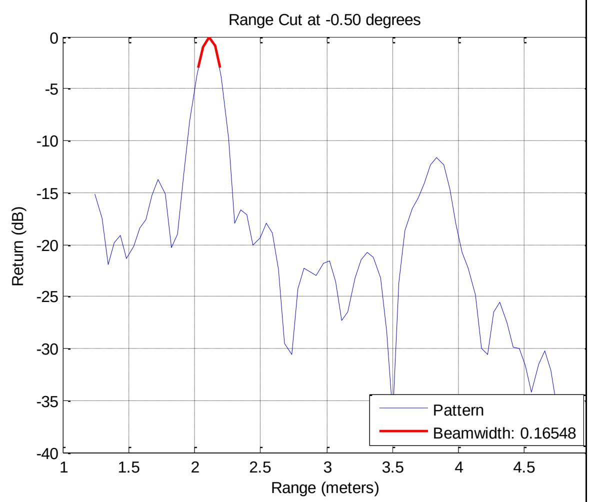


Figure 0-37 Range cut of second MIMO experiment with two targets

As before, we observe the angular resolution increase with just an addition of another transmitter. And like the other experiments in this task, we also observe sidelobe formation which is assumed to be due to improper element positioning.

Conclusion

The aim of these experiments were to prove the MIMO radar technique can be used to have better angular resolution with fixed aperture size or less number of elements for fixed angular resolution. The secondary goal of these experiments was to getting used to equipment and various digital signal processing techniques.

We have started with single-transmitter single-receiver radar case and slowly built-up to TDMA MIMO radar with the MISL's existing equipment. Even though we have faced many problems trying to adjust the equipment to our needs, we were able to use MIMO techniques with our arbitrary waveform generator and high sample rate oscilloscope. The initial problems have caused us a fixed delay when we started but we were able to pick up the pace and finished the experiments 6 weeks earlier than expected.

As can be observed from the results there are many flaws to be fixed in means of physical setup and software setup. But as a cumulative result, we have proven the MIMO technique in array radar context with real-life experiments. We have also gotten used to working with the anechoic test chamber, various types of antennas, amplifiers, filters, the AWG, the DPO, the cables, the signal processing techniques etc.

APPENDIX B

MIMO Simulator codes

```
% clear all
% close all
%% CONSTANTS
tools.constants

%% PARAMETERS
% WAVETYPE
WAVETYPE = ACOUSTIC_AIR;

% PHYSICS SETTINGS
TARGET_POSITIONS_TYPE = ABSOLUTE_TARGET_POSITIONS;

% FIGURE SETTINGS
if ~exist('FIGURE_NUMBER','var')
    FIGURE_NUMBER = 1;
end
FIGURE_NUMBER_START = FIGURE_NUMBER;

% REFERENCE WAVEFORM
REFERENCE_WAVEFORM_DIR = './waveforms/';
REFERENCE_WAVEFORM_FILENAME = 'lfm_38e3_42e3_250Ksps.txt';

FILE_EXTENSION = '.wfm';

% REFERENCE WINDOW
APPLY_MANUAL_WINDOW_TO_REFERENCE = TRUE;
REFERENCE_WINDOW_FUNCTION = @(x) gausswin(x, 2.5);
NORMALIZE_SAVED_DATA = FALSE;

% SIMULATION SETTINGS
NUM_OF_FRAMES = 2;
QUANTIZATION_SIMULATION = FALSE;
QUANTIZATION_ADC_BITS = 8;
DIRECT_PATH_TRANSMISSION = TRUE;
ENABLE_ANTENNA_SEPARATION_SHIFT = FALSE;

% MIN-MAXIMUMM RANGE
START_RANGE = FRAUNHOFER_DISTANCE; % meters
START_RANGE = 0.6; % meters DONT USE 'eps'!!
MAXIMUM_RANGE = 2.2; % meters
TARGET_THRESHOLD = sqrt(2)/2;
DYNAMIC_RANGE_DB = 20;

% BEAMFORMER ALGORITHM
BEAMFORMER_ALGORITHMS = [CONVENTIONAL_BEAMFORMER, NONLINEAR_BEAMFORMER,
BACKPROJECTION_BEAMFORMER, NONLINEAR_BACKPROJECTION, FFT_BEAMFORMER,
MUSIC_BEAMFORMER];
```

```
BEAMFORMER_ALGORITHMS = [NONLINEAR_BEAMFORMER];

% ARRAY TAPERING
ENABLE_ANGULAR_TAPERING = FALSE;
ANGULAR_TAPER_FUNCTION = @kaiser;

% NEARFIELD CORRECTION
ENABLE_NEARFIELD_CORRECTION = TRUE;

% INCREASED SAMPLE RATE VIA INTERP
ENABLE_INCREASED_SAMPLE_RATE = FALSE;
INCREASED_SAMPLE_RATE_RATIO = 100; % keep it less than 100

% MEASUREMENT PARAMS
if WAVETYPE == ELECTROMAGNETIC
    wavespeed = lightspeed;
elseif WAVETYPE == ACOUSTIC_AIR
    wavespeed = soundspeed_air;
elseif WAVETYPE == ACOUSTIC_UNDERWATER
    wavespeed = soundspeed_water;
end
PRI = 0.25;
rfFs = 250e3;
rfFstart = 38e3;
BW = 4e3;
rfFc = rfFstart + (BW/2);
lambda = wavespeed/rfFc;
k = 2*pi/(lambda);
analogFilterBw = 80e3;
lowIF = 38e3;
time_sample_count = ceil(MAXIMUM_RANGE/wavespeed*rfFs*2)+3000;
ENABLE_LOW_IF_CONVERSION = FALSE;
% OTHER PARAMS
Fs = rfFs;
Ts = 1/Fs;
if ENABLE_LOW_IF_CONVERSION
    lowIf = rfFstart;
end
lowIFc = lowIF+BW/2;
maxF = lowIF+BW;
prefilter_b = fir1(500, [lowIF/Fs*2 maxF/Fs*2],'DC-0');
prefilter_a = 1;
lofilter_b = fir1(500, BW/2/Fs*2,'low');
lofilter_a = 1;

% PLOT PARAMETERS
FIG_FILENAME_TEMPLATE =
'C:\Users\axs1106\Nextcloud\birmingham_docs\2017\8_aug\aperiodic_mimo_simulation_f
igures\'
SAVE_FIGURES = FALSE;
PLOT_INDIVIDUAL_SIGNALS = FALSE;
PLOT_MATCHED_SIGNALS = FALSE;

% TARGETS
targets = [
% % Experimental Scenario Targets
%              0.1 3.61 0.64+0.26       % PLATFORM FAR LEFT
%              0.235 3.645 0.64+0.26       % PLATFORM MID LEFT
```

```
%              0.32 3.67 0.64+0.26       % PLATFORM FAR LEFT
%              0.1 2.1 0.64+0.26       % WEIRD REFLECTION
%              0.5  1.64+0.6 0.64+0.15  % CLOSE LEFT
%              0.97 3.62 0.66+0.26         % PLATFORM RIGHT
%              1.08 3.62 0.66+0.26         % PLATFORM MID RIGHT
%              1.04 3.61875 0.66+0.26         % PLATFORM MID RIGHT 2
%              1.2 3.61 0.66+0.26         % PLATFORM FAR RIGHT
%              3.2 10.61 0.66+0.26         % 10m
% % 2D Plane Ideal Scenario Targets
%              12*lambda*sind(30)  12*lambda*cosd(30)   0.0
%              120*lambda*sind(-15)  120*lambda*cosd(-15)   0.0
%              120*lambda*sind(30)  120*lambda*cosd(30)   0.0
%              120*lambda*sind(15)  120*lambda*cosd(15)   0.0
%              120*lambda*sind(50)  120*lambda*cosd(50)   0.0
%              120*lambda*sind(70)  120*lambda*cosd(70)   0.0
%              120*lambda*sind(-50)  120*lambda*cosd(-50)   0.0
%              120*lambda*sind(-40)  120*lambda*cosd(-40)   0.0
%              120*lambda*sind(-30)  120*lambda*cosd(-30)   0.0
%              120*lambda*sind(0)  120*lambda*cosd(0)   0.0
%              150*sind(0)  150*cosd(0)   0.0
%              1*sind(45)  1*cosd(45)   0.0
%              0.99*sind(-0)  0.99*cosd(-0)   0.0
%              1.45*sind(-25)  1.45*cosd(-25)   0.0
%              1.4*sind(-6)  1.4*cosd(-6)   0.0
%              1.8*sind(-15)  1.8*cosd(-15)   0.0
            0.95*[sind(0)  cosd(0)   0.0]
            1.45*[sind(-25)  cosd(-25)   0.0]
            1.40*[sind(-6)  cosd(-6)   0.0]
% % Nearfield Scenario Targets
%                0.20+0.01  0.9+0.225 0.525+0.13 % 0 degree
%                0.20+0.44  0.75+0.225 0.525+0.13 % 25 degree
%                0.20+0.67  0.88+0.225 0.525+0.13 % 30 degree
%               0.20+1.05  0.8+0.225 0.525+0.13 % 45 degree
            ]; % x-y-z coordinates (meters)

targets_rcs = [
%                 0.075            % SMALL BALL
%                 0.025         % WEIRD REFLECTION
                4             % SOME BALL
                3.0             % NORMAL BALL
                1.75             % NORMAL BALL
%                 50.0            % 10m
               ];

% targets_rcs = ones(size(targets,1),1);
% ARRAY SETTINGS
NUM_TX_START = 1;
NUM_TX_END = 4;
NUM_RX_START = 1;
NUM_RX_END = 4;
NUM_TX = NUM_TX_END-NUM_TX_START+1; % DONT TOUCH
NUM_RX = NUM_RX_END-NUM_RX_START+1; % DONT TOUCH
rxElementSpacing = 0.5*lambda; %0.04; % meters;
% rxElementSpacing = 0.04; %0.04; % meters;
txElementSpacing = NUM_RX*rxElementSpacing; % meters
txElementSpacing = 4*rxElementSpacing; % meters

TX_ANTENNA_TYPE = @antennas.isotropic_antenna;
RX_ANTENNA_TYPE = @antennas.isotropic_antenna;
```

```
% GEOMETRY
RECEIVE_ARRAY_LOCATION = RELATIVE_MIDDLE;
ROOM_SIZE_X = 5*0.6;
ROOM_SIZE_Y = 9*0.6;
ROOM_SIZE_Z = 5*0.6;
PRISM_DRAW_ROUTE = [0 1 5 1 3 7 3 2 6 7 5 4 0 2 6 4];
ROOM_BORDERS = (dec2bin(PRISM_DRAW_ROUTE)-
'0').*(ones(length(PRISM_DRAW_ROUTE),1)*[ROOM_SIZE_X ROOM_SIZE_Y ROOM_SIZE_Z]);
tx_antennas = [
% % Experimental Scenario Antennas
%                 0.52 0.20 0.86+0.26
%                 0.72 0.20 0.86+0.26
%                 0.92 0.20 0.86+0.26
% % Nearfield Scenario Antennas
%                 0+0*txElementSpacing  0   0.825
%                 0+1*txElementSpacing  0   0.825
%                 0+2*txElementSpacing  0   0.825
%                 0+3*txElementSpacing  0   0.825
% % 2D Plane Ideal Scenario Antennas
%                 0+0*txElementSpacing  0   0.65
%                 0+1*txElementSpacing  0   0.65
%                 0+2*txElementSpacing  0   0.65
%                 0+3*txElementSpacing  0   0.65
% % Minimum Redundancy Thinned Mimo
%                 0+0*lambda/2  0   0
%                 0+1*lambda/2  0   0
%                 0+9*lambda/2  0   0
%                 0+12*lambda/2  0   0
% % Nonlinear array
%                 0+0*lambda  0   0.65
%                 0+1.8*lambda  0   0.65
%                 0+4.2*lambda  0   0.65
%                 0+6.75*lambda  0   0.65
% % 3D Printed nonlinear array
              0+0*lambda  0   0.66
               0+2.4750*lambda  0   0.66
               0+4.8044*lambda  0   0.66
               0+6.5515*lambda  0   0.66
                ]+(0.0.*lambda*rand(4,3)-0.05);
rx_antennas = [
% % Experimental Scenario Antennas
%                  0.52 0.53 0.48
%                  0.56 0.53 0.48
%                  0.60 0.53 0.48
%                  0.64 0.53 0.48
%                  0.68 0.53 0.48
% % Nearfield Scenario Antennas
%                 0+0*rxElementSpacing  0.225   0.21
%                 0+1*rxElementSpacing  0.225   0.21
%                 0+2*rxElementSpacing  0.225   0.21
%                 0+3*rxElementSpacing  0.225   0.21
%                 0+4*rxElementSpacing  0.225   0.21
% % 2D Plane Ideal Scenario Antennas
%                 0+0*rxElementSpacing  0.04   0.60
%                 0+1*rxElementSpacing  0.04   0.60+1*11.5e-3
%                 0+2*rxElementSpacing  0.04   0.60+2*11.5e-3
%                 0+3*rxElementSpacing  0.04   0.60+3*11.5e-3
%                 0+4*rxElementSpacing  0   0.60
```

```
% % Minimum Redundancy Thinned Mimo
%                0+0*lambda/2  0   0
%                0+1*lambda/2  0   0
%                0+9*lambda/2  0   0
%                0+12*lambda/2  0   0
% % Nonlinear array
%                0+0*lambda  0.04   0.6+0*12.2e-3
%                0+0.5*lambda  0.04   0.6+1*12.2e-3
%                0+1.05*lambda  0.04   0.6+2*12.2e-3
%                0+6.3*lambda  0.04   0.6+3*12.2e-3
% % 3D Printed nonlinear array
              0+0*lambda  0.0   0.6
              0+5.0959*lambda  0.0   0.6
              0+5.6294*lambda  0.0   0.6
              0+6.1147*lambda  0.0   0.6
               ]+(0.0.*lambda*rand(4,3)-0.05);

% tx_antennas = flip(tx_antennas,1);
% rx_antennas = flip(rx_antennas,1);

% Dong, Liu, Jiang, Hu, Shi - MR-MIMO Using Cyclic Permutation of Perfect Distance
% 6 + 12
% tx_array_positions = [0 1 8 11 13 17] *lambda/2;
% rx_array_positions = [0, 1*31, 3*31, 6*31, 13*31, 20*31, 27*31, 34*31, 41*31, 
45*31, 49*31, 50*31] *lambda/2;
% tx_antennas = [tx_array_positions' zeros(6,1) zeros(6,1)];
% rx_antennas = [rx_array_positions' zeros(12,1) zeros(12,1)];

% He, Feng, Younan, Li - Thinned MIMO using multiple Genetic Algorithm
% 15+16
% Table 1 - After 2st Optimization
% tx_array_positions = [0 0.5625 1.9375 2.5625 3.6250 4.9688 5.6250 6.6875 7.3750 
9.5156 10.3281 11.00 11.5625 13.3125 14.00] *lambda/1;
% rx_array_positions = [0 14.9688 29.5938 44.8594 59.7344 74.6875 89.4063 105.2656 
120.3750 134.8125 150.50 165.1406 179.5156 194.9688 210.00 225.00] *lambda/1;
% tx_antennas = [tx_array_positions' zeros(15,1) zeros(15,1)];
% rx_antennas = [rx_array_positions' zeros(16,1) zeros(16,1)];

% PLATFORM OFFSET
% tx_antennas = tx_antennas + ones(size(tx_antennas,1),1)*[0 0.6 0];
% rx_antennas = rx_antennas + ones(size(rx_antennas,1),1)*[0 0.6 0];
% targets = targets + ones(size(targets,1),1)*[0 0.6 0];
% ALL POSITIONS RE-ALIGNED TO PHYSICAL CENTER
physical_center = [(tx_antennas(1,X)+tx_antennas(end,X))/2, 
(tx_antennas(1,Y)+tx_antennas(end,Y))/2, (tx_antennas(1,Z)+tx_antennas(end,Z))/2];
% Target positions should stay relativeto 0,0,0 point UNLESSS experimental
% positions are simulated. Hence, the new parameter:
if TARGET_POSITIONS_TYPE == RELATIVE_TARGET_POSITIONS
    targets = targets - ones(size(targets,1),1)*physical_center;
end
tx_antennas = tx_antennas - ones(size(tx_antennas,1),1)*physical_center;
rx_antennas = rx_antennas - ones(size(rx_antennas,1),1)*physical_center;
% ARRAY LOCATION OFFSET
if RECEIVE_ARRAY_LOCATION == RELATIVE_MIDDLE
    rx_antennas = rx_antennas + 
ones(size(rx_antennas,1),1)*[(tx_antennas(NUM_TX,X)/2-tx_antennas(1,X)/2)-
(rx_antennas(NUM_RX,X)/2-rx_antennas(1,X)/2) 0 0];
elseif RECEIVE_ARRAY_LOCATION == RELATIVE_LEFT
```

```
    throw(MException('MIMO_SIMULATOR:NotYetImplemented','This functionality is not
yet implemented'))
elseif RECEIVE_ARRAY_LOCATION == RELATIVE_RIGHT
    throw(MException('MIMO_SIMULATOR:NotYetImplemented','This functionality is not
yet implemented'))
end
physical_center = zeros(1,3);

%% ASSERTIONS
% Not yet implemented
assert( strcmpi(func2str(TX_ANTENNA_TYPE),'antennas.isotropic_antenna') &&
strcmpi(func2str(RX_ANTENNA_TYPE),'antennas.isotropic_antenna'), 'Stick with
isotropic antennas until we know better.\n');
% Safety
assert(INCREASED_SAMPLE_RATE_RATIO <= 100, 'Higher increased sample ratios kill
the cpu!\n');
assert(NUM_TX_START == 1 && NUM_RX_START == 1, 'MIMO radar simulator cannot yet
simulate individual elements.\n');
assert(MAXIMUM_RANGE >= max(targets(:,Y)), 'Maximum range is not big enough to
cover furthest target.\n');
assert(size(tx_antennas,1) >= NUM_TX,'Number of tx antenna positions do not match
the number of tx antennas.\n')
assert(size(rx_antennas,1) >= NUM_RX,'Number of rx antenna positions do not match
the number of rx antennas.\n')

same_placed_antenna = false;
for t=NUM_TX_START:NUM_TX_END
    for r=NUM_RX_START:NUM_RX_END
        if tx_antennas(t,:) == rx_antennas(r,:)
            same_placed_antenna = true;
        end
    end
end
if DIRECT_PATH_TRANSMISSION == TRUE
    assert( same_placed_antenna == false,'if direct path transmission is going to
be simulated, antennas cant be placed in physically same locations!\n')
end
choice = questdlg('Should I RESET the data and CLOSE the open figures?', 'RESET
Data and CLOSE Figures?','Yes','No','Yes');
switch choice
    case 'Yes'
        close all
        clear sig frames fnames ref* ref_bb *beamform* *steered* *bb* displayim
*taper*
        FIGURE_NUMBER_START = 1;
    case 'No'
        choice = choice;
        FIGURE_NUMBER_START = FIGURE_NUMBER;
end
%% PLOT SCENARIO
radar_simulators.headless_scenario_sketcher

%% REFERENCE EXTRACTION
[ref, ~] = tools.import_signal([REFERENCE_WAVEFORM_DIR
REFERENCE_WAVEFORM_FILENAME], rfFs);
%%%%%%%%%%%%%%%%%%%
% TODO: MIGHT BE USEFUL
orig_ref = ref;
if APPLY_MANUAL_WINDOW_TO_REFERENCE == TRUE
```

```
    ref = ref.*REFERENCE_WINDOW_FUNCTION(length(ref));
end
%%%%%%%%%%%%%%%%%
recordlen = time_sample_count + length(ref);
t = ((0:(recordlen-1))/rfFs)';
ref = padarray(ref, recordlen - length(ref), 0 , 'post');
orig_ref = padarray(orig_ref, recordlen - length(orig_ref), 0 , 'post');
ref_sig = orig_ref;
if ENABLE_LOW_IF_CONVERSION
    ref = ref.*sin(2*pi*(rfFstart-lowIF)*t);
    ref = filter(fir1(400, analogFilterBw/rfFs*2), 1, ref);
end
ref = resample(ref, round(Fs), rfFs, 50);
ref = filter(prefilter_b,prefilter_a,ref);
ref = padarray(ref, recordlen-length(ref) , 0, 'post');
tref = ((0:size(ref,1)-1)/Fs)';
ref_bb = ref.*exp(-2j*pi*lowIFc*tref);
ref_bb = filter(lofilter_b, lofilter_a, ref_bb, [], 1);

%% SIMULATION

if ENABLE_INCREASED_SAMPLE_RATE == TRUE
    processingFs = INCREASED_SAMPLE_RATE_RATIO*Fs;
else
    processingFs = Fs;
end
processingTs = 1/processingFs;

measurement_data = zeros(recordlen, NUM_RX, NUM_TX);
matched_measurement_data = zeros(recordlen, NUM_RX, NUM_TX);
for tx_index = NUM_TX_START:NUM_TX_END
    tx_antenna = tx_antennas(tx_index,:);
    for rx_index = NUM_RX_START:NUM_RX_END
        rx_antenna = rx_antennas(rx_index,:);

        % DIRECT PATH TRANSMISSION SIMULATION
        if DIRECT_PATH_TRANSMISSION == TRUE
            tx2rx = pdist2(tx_antenna, rx_antenna, 'euclidean');
            tx2rx_az = atand((rx_antenna(X)-tx_antenna(X))/(rx_antenna(Y)-
tx_antenna(Y)));
            tx2rx_ev = atand((rx_antenna(Z)-tx_antenna(Z))/(rx_antenna(Y)-
tx_antenna(Y)));
            rx2tx_az = atand((tx_antenna(X)-rx_antenna(X))/(tx_antenna(Y)-
rx_antenna(Y)));
            rx2tx_ev = atand((tx_antenna(Z)-rx_antenna(Z))/(tx_antenna(Y)-
rx_antenna(Y)));
            % Copy original waveform
            sig = ref_sig;
            % Time delay
            sig = delayseq(sig, tx2rx/wavespeed, rfFs);
            % Power loss
            sig = sig/(tx2rx.^2);
            % Antenna Loss
            sig = sig.*TX_ANTENNA_TYPE( tx2rx_ev, tx2rx_az);
            sig = sig.*RX_ANTENNA_TYPE( rx2tx_ev, rx2tx_az);
            measurement_data(:,rx_index-NUM_RX_START+1,tx_index-NUM_TX_START+1) =
measurement_data(:,rx_index-NUM_RX_START+1,tx_index-NUM_TX_START+1) + sig;
        end
```

```
        % TARGET SIMULATION
        for target_index = 1:size(targets,1)
            target = targets(target_index, :);
            target_rcs = targets_rcs(target_index);
            tx2target = pdist2(tx_antenna, target, 'euclidean');
            tx2target_az = atand((target(X)-tx_antenna(X))/(target(Y)-
tx_antenna(Y)));
            tx2target_ev = atand((target(Z)-tx_antenna(Z))/(target(Y)-
tx_antenna(Y)));

            target2rx = pdist2(target, rx_antenna, 'euclidean');
            target2rx_az = atand((target(X)-rx_antenna(X))/(target(Y)-
rx_antenna(Y)));
            target2rx_ev = atand((target(Z)-rx_antenna(Z))/(target(Y)-
rx_antenna(Y)));
            % Copy original waveform
            sig = ref_sig;
            % Time delay
            sig = delayseq(sig, (tx2target+target2rx)/wavespeed, rfFs);
            % Target RCS
            sig = target_rcs*sig;
            % Power loss
            sig = sig/sqrt((tx2target.^2)/(target2rx.^2)); % sqrt because signal
amplitude not power
            % Antenna Loss
            sig = sig.*TX_ANTENNA_TYPE( tx2target_ev, tx2target_az);
            sig = sig.*RX_ANTENNA_TYPE( target2rx_ev, target2rx_az);

            measurement_data(:,rx_index-NUM_RX_START+1,tx_index-NUM_TX_START+1) =
measurement_data(:,rx_index-NUM_RX_START+1,tx_index-NUM_TX_START+1) + sig;
        end

        % ANALOG CIRCUITRY and DPO SIMULATION
        data_bb = measurement_data(:,rx_index-NUM_RX_START+1,tx_index-
NUM_TX_START+1);
        data_bb = awgn(data_bb, mag2db(NUM_OF_FRAMES), 'measured');
        % analog mixer & filter
        if ENABLE_LOW_IF_CONVERSION
            data_bb = data_bb.*sin(2*pi*(rfFstart-lowIF)*t);
            data_bb = filter(fir1(400, analogFilterBw/rfFs*2), 1, data_bb);
        end
        % dpo capture
        data_bb = resample(data_bb, round(Fs), rfFs, 50);
        data_bb = padarray(data_bb, recordlen-length(data_bb) , 0, 'post');
        if QUANTIZATION_SIMULATION == TRUE
            data_bb = quantiz(data_bb, linspace(0,1,pow2(QUANTIZATION_ADC_BITS)-
1));
        end
        % processing
        data_bb = data_bb - repmat(mean(data_bb ,1), size(data_bb,2), 1);
        data_bb = filter(prefilter_b,prefilter_a,data_bb, [], 1);
        data_bb = data_bb.*repmat(exp(-2j*pi*lowIFc*tref), 1, size(data_bb,2));
        data_bb = filter(lofilter_b, lofilter_a, data_bb, [], 1);

        data_bbcorr = (ifft(fft(data_bb, recordlen, 1).*fft(flipud(conj(ref_bb)),
recordlen, 1)));
```

```
        if ENABLE_ANTENNA_SEPARATION_SHIFT == TRUE && DIRECT_PATH_TRANSMISSION ==
TRUE
            [pks, locs] = findpeaks(abs(data_bbcorr),'SORTSTR','descend');
            peakindex(1) = locs(1);

            separationShift = round(tx2rx/wavespeed/processingTs);
            data_bbcorr = delayseq(data_bbcorr, -locs(1)+separationShift);
            data_bb = delayseq(data_bb, -locs(1)+separationShift);
        end

        measurement_data(:,rx_index-NUM_RX_START+1,tx_index-NUM_TX_START+1) =
data_bb;
        matched_measurement_data(:,rx_index-NUM_RX_START+1,tx_index-
NUM_TX_START+1) = data_bbcorr;

        if NORMALIZE_SAVED_DATA == TRUE
            maxVal = max(abs(matched_measurement_data(:,rx_index-
NUM_RX_START+1,tx_index-NUM_TX_START+1)));
            matched_measurement_data(:,rx_index-NUM_RX_START+1,tx_index-
NUM_TX_START+1) =  matched_measurement_data(:,rx_index-NUM_RX_START+1,tx_index-
NUM_TX_START+1)/maxVal;
        end
    end
end

%% SIGNAL PLOTS
if PLOT_INDIVIDUAL_SIGNALS
    fig = figure(FIGURE_NUMBER);
    close(fig)
    fig = figure(FIGURE_NUMBER);
    FIGURE_NUMBER = FIGURE_NUMBER + 1;
    hold all
    grid on
    for tx_index = NUM_TX_START:NUM_TX_END
        tx_antenna = tx_antennas(tx_index,:);
        for rx_index = NUM_RX_START:NUM_RX_END
            rx_antenna = rx_antennas(rx_index,:);
            data =
((measurement_data(1:ceil(MAXIMUM_RANGE/wavespeed*Fs*2),rx_index,tx_index)));
            data = (data);

plot((tref(1:ceil(MAXIMUM_RANGE/wavespeed*Fs*2))*wavespeed/2),real(data))

plot((tref(1:ceil(MAXIMUM_RANGE/wavespeed*Fs*2))*wavespeed/2),imag(data))
            set(gca,'FontSize',14,'fontWeight','bold')

    %         ylim([-60 0])
            [lgh,~,~,~] = legend;
            existing_legends = get(lgh,'String');
            existing_legends{size(existing_legends,2)+1} =
sprintf('Real(tx%d_rx%d)',tx_index,rx_index);
            existing_legends{size(existing_legends,2)+1} =
sprintf('Imag(tx%d_rx%d)',tx_index,rx_index);
            legend(existing_legends,'Interpreter','none')
            title(sprintf('Baseband'))
        end
    end
%     ylim([-60 60])
```

```
end

if PLOT_MATCHED_SIGNALS
    fig = figure(FIGURE_NUMBER);
    close(fig)
    fig = figure(FIGURE_NUMBER);
    FIGURE_NUMBER = FIGURE_NUMBER + 1;
    hold all
    grid on
    for tx_index = NUM_TX_START:NUM_TX_END
        tx_antenna = tx_antennas(tx_index,:);
        for rx_index = NUM_RX_START:NUM_RX_END
            rx_antenna = rx_antennas(rx_index,:);
            data =
(abs(matched_measurement_data(1:ceil(MAXIMUM_RANGE/wavespeed*Fs*2),rx_index,tx_ind
ex)));
            data = mag2db(data);
            plot((tref(1:ceil(MAXIMUM_RANGE/wavespeed*Fs*2))*wavespeed/2),data)
            set(gca,'FontSize',14,'fontWeight','bold')

    %         ylim([-60 0])
            [lgh,~,~,~] = legend;
            existing_legends = get(lgh,'String');
            existing_legends{size(existing_legends,2)+1} =
sprintf('tx%d_rx%d',tx_index,rx_index);
            legend(existing_legends,'Interpreter','none')
            title(sprintf('Matched'))
        end
    end
%     ylim([-60 60])
end

if START_RANGE == 0
    START_RANGE = eps;
elseif START_RANGE == FRAUNHOFER_DISTANCE
    START_RANGE = 2*((NUM_TX*txElementSpacing)^2)/(lambda)*0.9; % simulations
showed that 80% of fraunhofer distance is still OK
end
rangerange = ceil(MAXIMUM_RANGE/wavespeed/Ts*2):-
1:ceil(START_RANGE/wavespeed/processingTs*2);

%% BEAMFORMER PROCESS
if ~isempty(find(BEAMFORMER_ALGORITHMS == CONVENTIONAL_BEAMFORMER, 1))
    scan_angles =  -89.5:.5:89.5;

    if START_RANGE == 0
        START_RANGE = eps;
    elseif START_RANGE == FRAUNHOFER_DISTANCE
        START_RANGE = 2*((NUM_TX*txElementSpacing)^2)/(lambda)*0.9; % simulations
showed that 80% of fraunhofer distance is still OK
    end
    rangerange = ceil(MAXIMUM_RANGE/wavespeed/Ts*2):-
1:ceil(START_RANGE/wavespeed/processingTs*2);
    range_vector = rangerange*processingTs*wavespeed/2;

    steered_beams = beamformers.conventional_beamformer(scan_angles, range_vector,
matched_measurement_data(rangerange,:,:), NUM_TX_START:NUM_TX_END,
```

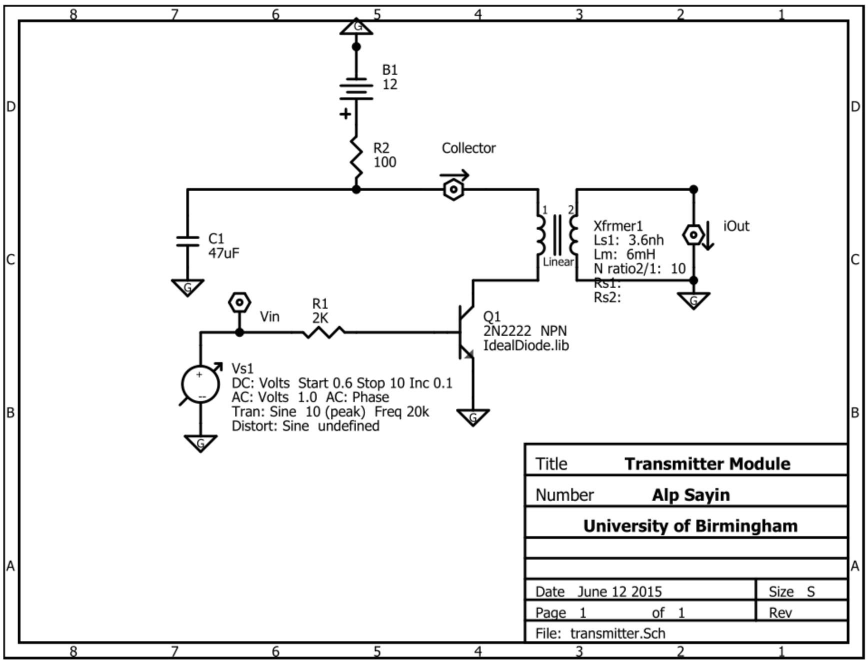
```
NUM_RX_START:NUM_RX_END, txElementSpacing, rxElementSpacing, lambda,
ENABLE_NEARFIELD_CORRECTION, ENABLE_ANGULAR_TAPERING, ANGULAR_TAPER_FUNCTION);

    displayim = abs(steered_beams);
    displayim = mag2db(displayim/max(max(displayim)));

    [fig, FIGURE_NUMBER] = tools.plot_range_azimuth_map(FIGURE_NUMBER,
range_vector, scan_angles, displayim, DYNAMIC_RANGE_DB, ...
        sprintf('Range-Azimuth Map (Conventional Beamformer) [%s]',scenario) ...
        );
end
%% APERIODIC BEAMFORMER PROCESS
if ~isempty(find(BEAMFORMER_ALGORITHMS == NONLINEAR_BEAMFORMER, 1))
    scan_angles =  -89.5:.5:89.5;
    tx_array_positions = tx_antennas(:,X);
    rx_array_positions = rx_antennas(:,X);

    if START_RANGE == 0
        START_RANGE = eps;
    elseif START_RANGE == FRAUNHOFER_DISTANCE
        START_RANGE = 2*((NUM_TX*txElementSpacing)^2)/(lambda)*0.9; % simulations
showed that 80% of fraunhofer distance is still OK
    end
    rangerange = ceil(MAXIMUM_RANGE/wavespeed/Ts*2):-
1:ceil(START_RANGE/wavespeed/processingTs*2);
    range_vector = rangerange*processingTs*wavespeed/2;

    steered_beams = beamformers.nonlinear_beamformer(scan_angles, range_vector,
matched_measurement_data(rangerange,:,:),
tx_array_positions(NUM_TX_START:NUM_TX_END),
rx_array_positions(NUM_RX_START:NUM_RX_END), lambda, ENABLE_NEARFIELD_CORRECTION);

    displayim = abs(steered_beams);
    displayim = mag2db(displayim/max(max(displayim)));

    [figs, FIGURE_NUMBER] = tools.plot_range_azimuth_map(FIGURE_NUMBER,
range_vector, scan_angles, displayim, DYNAMIC_RANGE_DB, ...
        sprintf('Range-Azimuth Map (Nonlinear Beamformer) [%s]',scenario) ...
        );

    tools.conditional_savefig(SAVE_FIGURES, figs(1),
strcat(FIG_FILENAME_TEMPLATE,figs(1).Children(2).Title.String,'.fig'),'compact');
    tools.conditional_savefig(SAVE_FIGURES, figs(2),
strcat(FIG_FILENAME_TEMPLATE,figs(2).Children(2).Title.String,'.fig'),'compact');
    tools.conditional_savefig(SAVE_FIGURES, figs(3),
strcat(FIG_FILENAME_TEMPLATE,figs(3).Children(2).Title.String,'.fig'),'compact');

%     [figs, FIGURE_NUMBER] = tools.plot_range_crossrange_map(FIGURE_NUMBER,
range_vector, scan_angles, displayim, DYNAMIC_RANGE_DB, ...
%         sprintf('Range-CrossRange Map (Nonlinear Beamformer) [%s]', scenario)
...
%         );
%     close(figs(2))
%     close(figs(3))
%     figure(figs(1))
%     h = [   -MAXIMUM_RANGE   MAXIMUM_RANGE  0  iif(MAXIMUM_RANGE>5,
MAXIMUM_RANGE-5, inline_else, MAXIMUM_RANGE) ];
%     axis(h);
```

```
    tools.conditional_savefig(SAVE_FIGURES, figs(1), 
strcat(FIG_FILENAME_TEMPLATE,figs(1).Children(2).Title.String,'.fig'),'compact');
end


%% HOGBOM CLEAN PREPARATION

ANGLE_SAMPLESET = scan_angles;
params.lambda = lambda
params.tx_antennas = tx_antennas
params.rx_antennas = rx_antennas
```

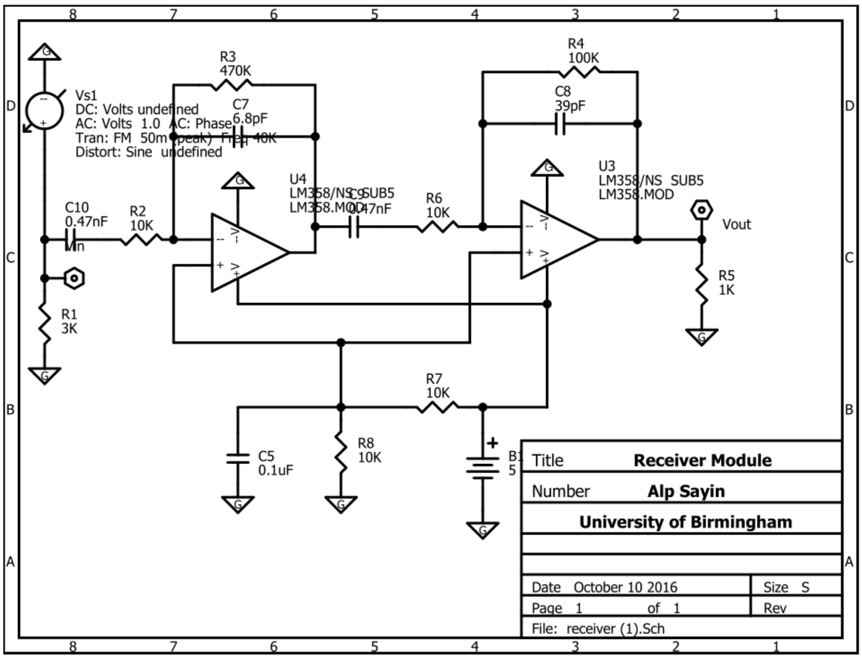
APPENDIX C

Acoustic MIMO Technology Demonstrator Implementation

Transmitter module circuit schematic

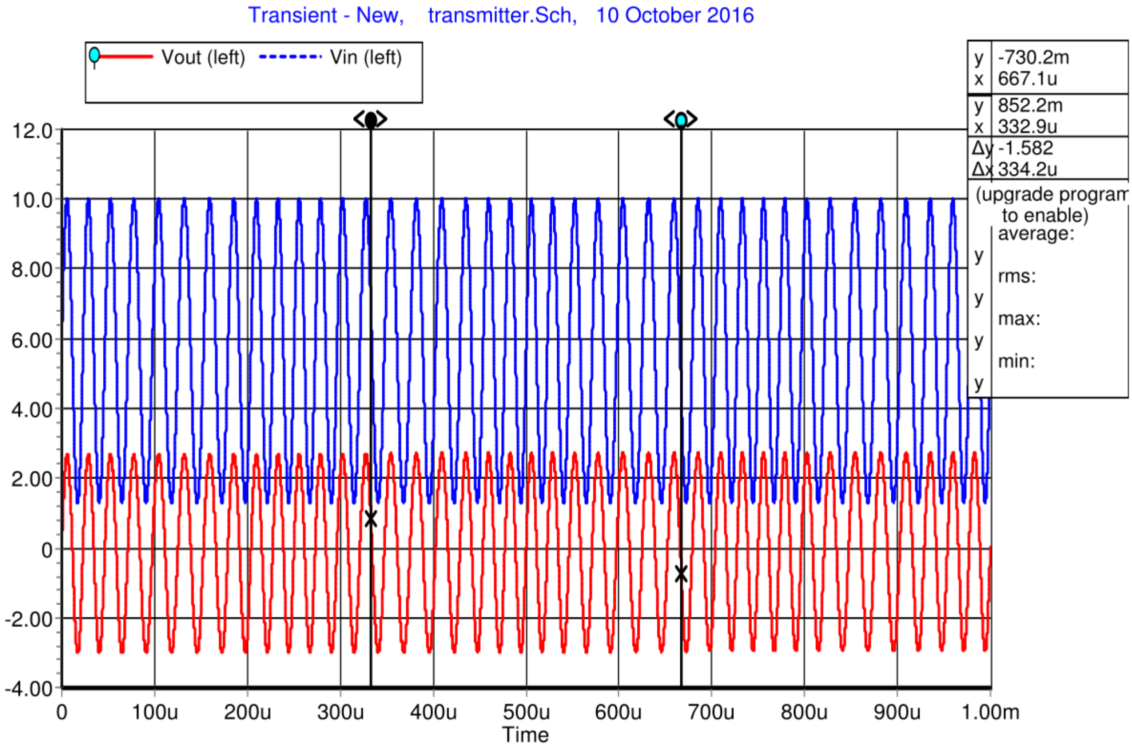


Receiver Module Schematic

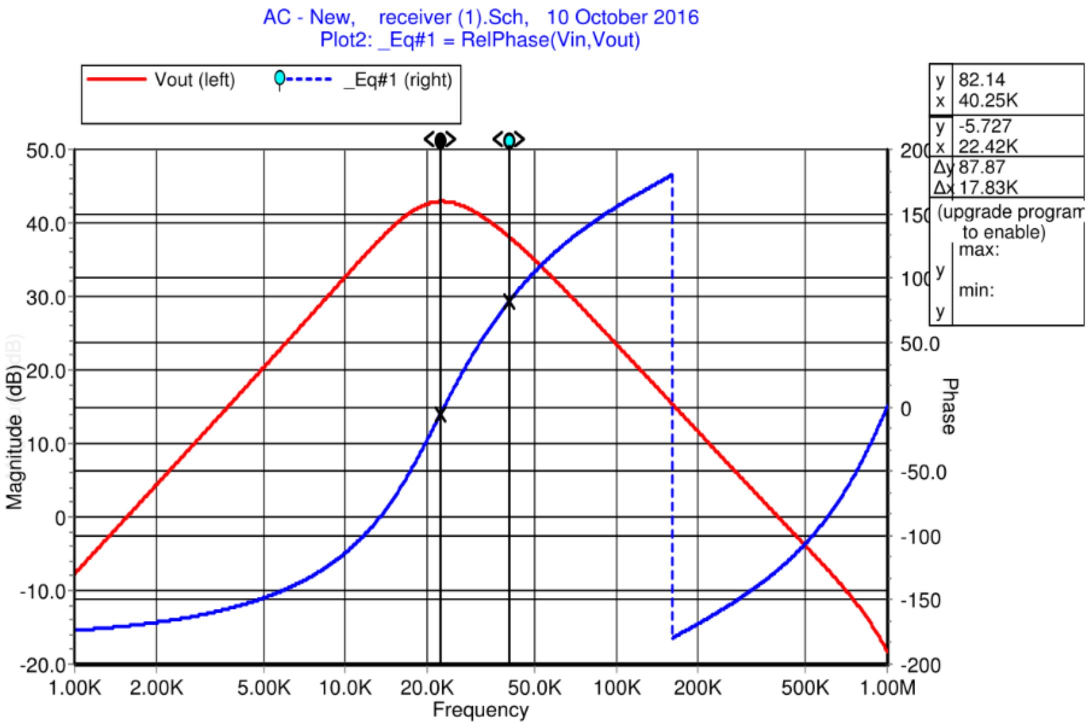


APPENDIX C

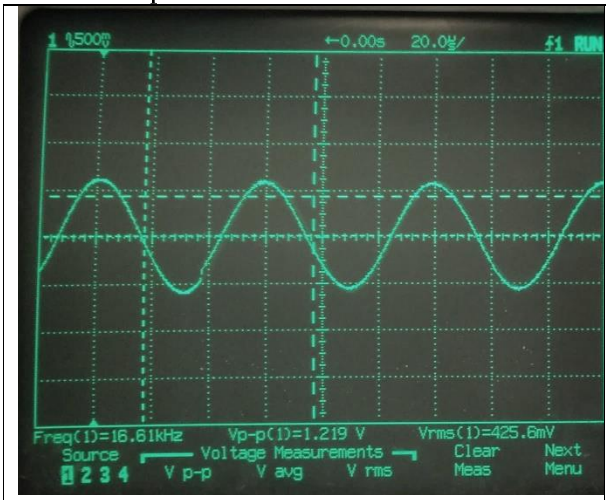
5Spice simulation of transmitter module (transient analysis have been used to overcome transistor being simulated

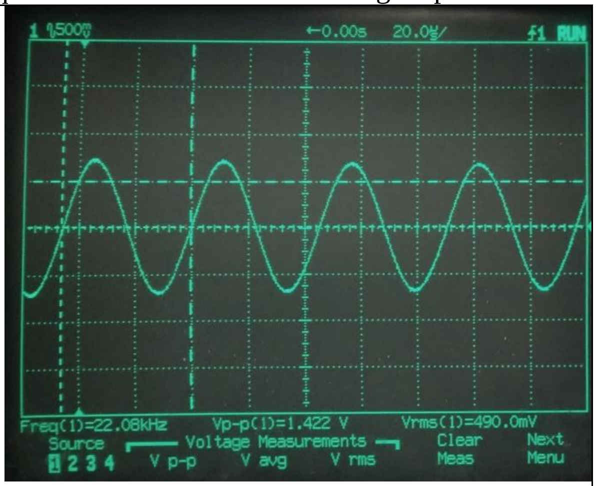


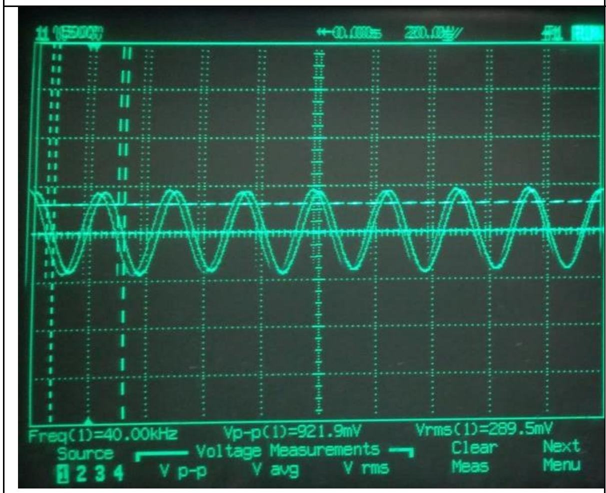
5Spice simulation of receiver module

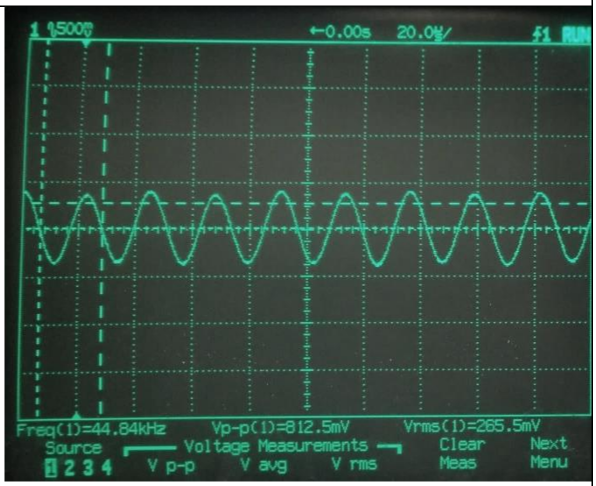


Receiver amplification measurements at various frequencies with shown RMS voltage input









Simulated and measured amplification levels

Frequency	Simulated Amplification	Measured Amplification
16.6 kHz	110.6 (20.4 dB)	121.6 (20.8 dB)
22.1 kHz	139.6 (21.4 dB)	140.0 (21.5 dB)
40.0 kHz	80.19 (19.0 dB)	82.7 (19.2 dB)
44.8 kHz	67.53 (18.3 dB)	75.9 (18.8 dB)

Input Settings of the Frequency Generator used to test the receiver amplification levels

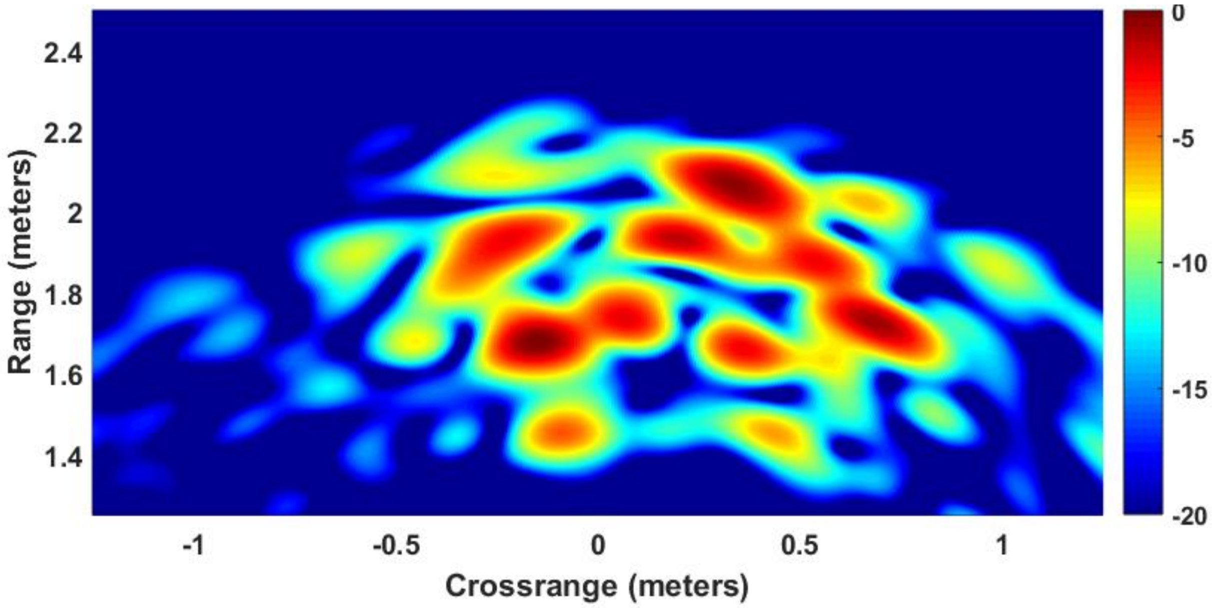
APPENDIX C


Black Star
LDO 100 LOW DISTORTION SINE/SQUARE OSCILLATOR
22.8KHz Sine
3.5mV rms
OUTPUT
(600 Ω)
FREQUENCY
RANGE (Hz)
100 1K 10K 100K
AMPLITUDE
ATTENUATOR (dB)
SINE 0 −10 −20 −30 −40
SQUARE

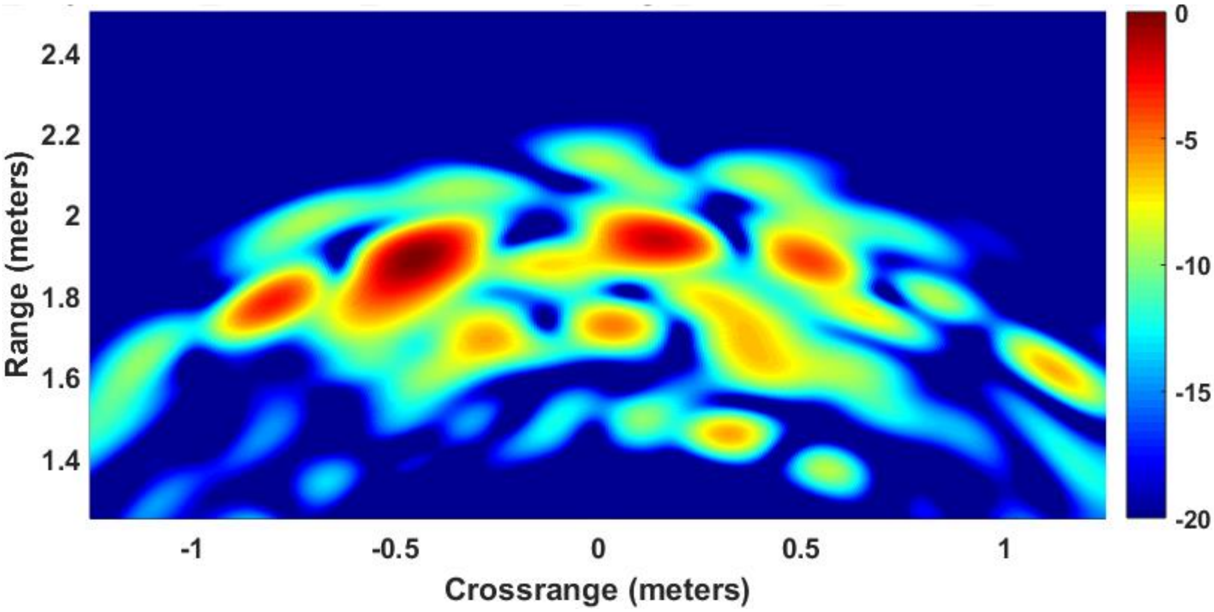
APPENDIX D

7.4.0 Intermediary results from experiments with the mountain bicycle

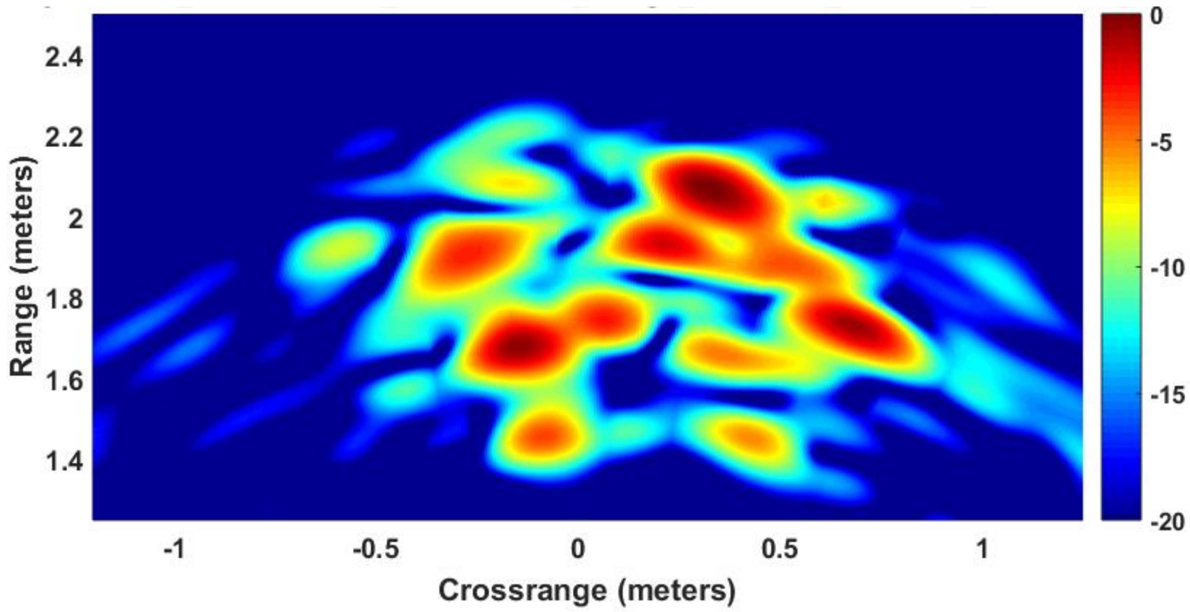
Non-CLEANed Range-crossrange map of mountain bike obtained via experiments with conventional MIMO sensor array



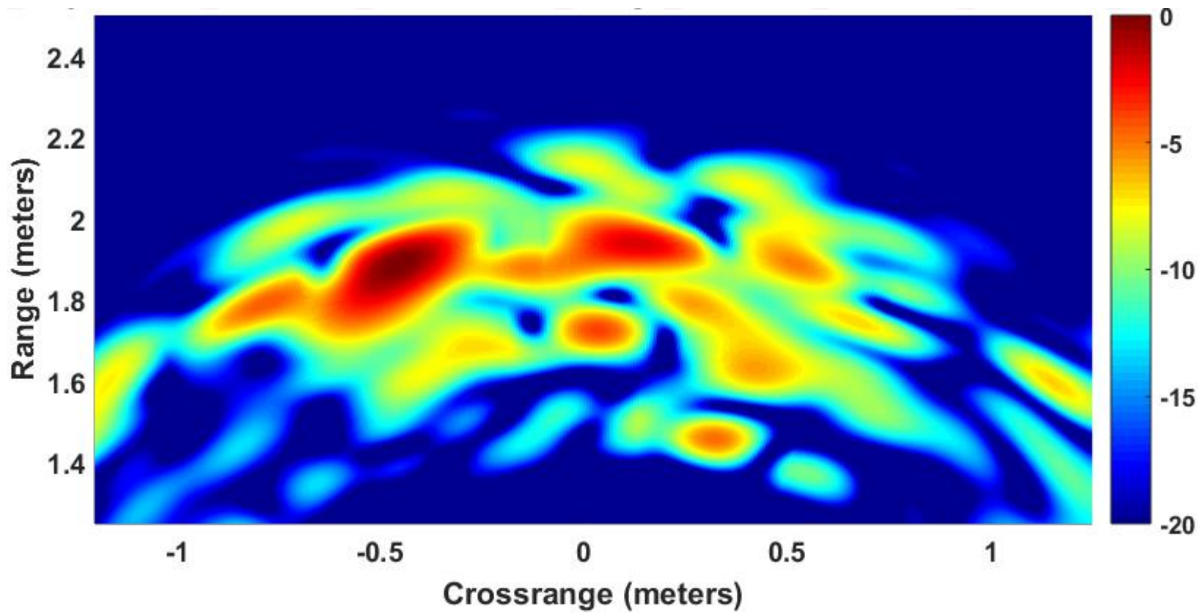
Non-CLEANed Range-crossrange map of mountain bike obtained via experiments with aperiodic MIMO sensor array



CLEANed Range-crossrange map of mountain bike obtained via experiments with conventional MIMO sensor array



CLEANed Range-crossrange map of mountain bike obtained via experiments with aperiodic MIMO sensor array



APPENDIX E

7.4.1 Code segment summarising the range/angle dependent platform motion compensation

```
slowtime_t = permute((0:size(steered_slowtime_frames,3)-1)'*PRI, [2 3 1]);
[azimuth_mesh, range_mesh, slowtime_mesh] = meshgrid(scan_angles, range_vector,
(slowtime_t));
[aircraft_velocity_az, aircraft_velocity_el, aircraft_velocity_r] =
cart2sph(AIRCRAFT_VELOCITY(X), AIRCRAFT_VELOCITY(Y), AIRCRAFT_VELOCITY(Z));
relative_aircraft_velocity_az = rad2deg(aircraft_velocity_az)-azimuth_mesh;
% compute the relative vector component that shoots straight through the target
aircraft_doppler_frequency =
2*aircraft_velocity_r./lambda.*cosd(relative_aircraft_velocity_az);
phase_shifter = exp(-2j*pi*aircraft_doppler_frequency.*slowtime_mesh);
steered_slowtime_frames_prefft = steered_slowtime_frames_prefft.*phase_shifter;
```