DAT096
Ultrasonic Sound Localization using Microphone Array Beamforming

Group M3

John Croft (presenting)
Hezhe Xiao

Ashwin Kumar Balakrishnan
Prema Manickavasagam

Agenda

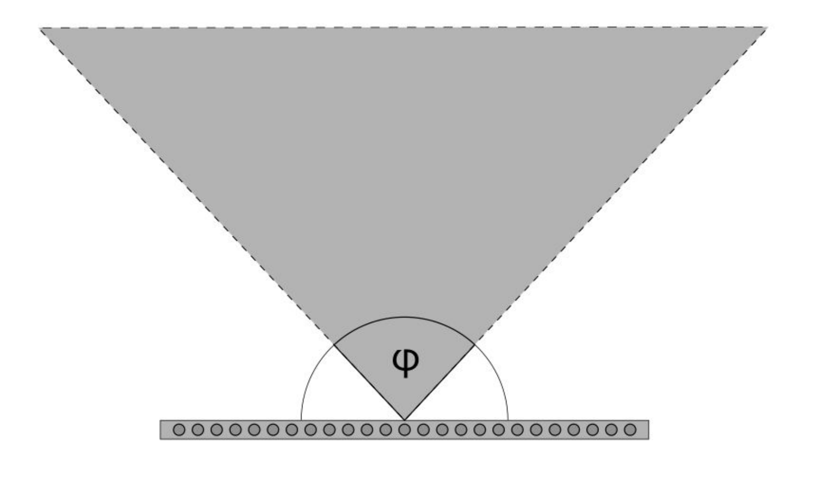
- Review project goals, specifications & limitations
- Recap beamforming theory
- Hardware & Interfacing
- Design Implementation
- Results
- Conclusions
- Future Work

Introduction

Goals

- Locate and track a moving ultrasonic audio source using *beamforming*.
- Implement system on an Artix-7 FPGA equipped with linear microphone arrays.
- Adhere to SCRUM principles throughout project.

Specifications



- Locate audio source in two dimensions.
- Determine direction of source to within 2° in the cone of interest.
- Detect changes in location with a latency of <1s.
- Display direction of audio source in real time.

Limitations

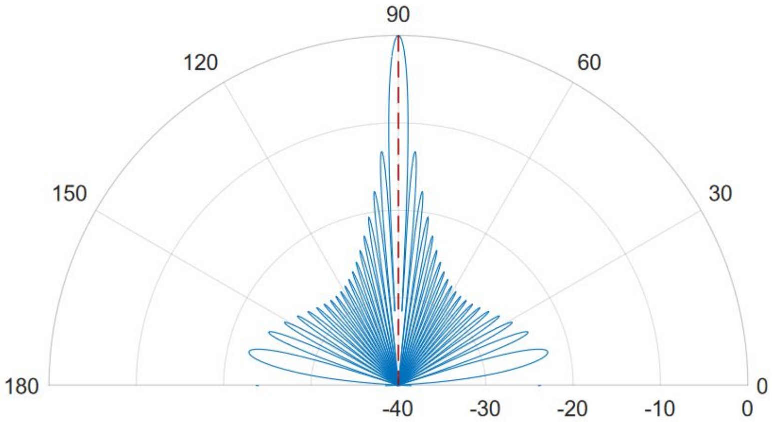
- Beamforming can be accomplished using many different techniques of varying complexity
  - We use the simplest variant: delay-and-sum (DAS).
- Due to *unfortunate circumstances,* only one sensor array was available, constraining the system to one dimension.

Background Theory

Delay-and-Sum Beamforming

Beamforming is a spatial filtering technique in which an array of *omnidirectional* sensors approximates a *directional* sensor.

The beamforming filter-response can be represented as a **beampattern** (right), with gain vs. incident angle of signal.



Theoretical beampattern modelled in MATLAB. 24 sensors, 25 kHz source signal

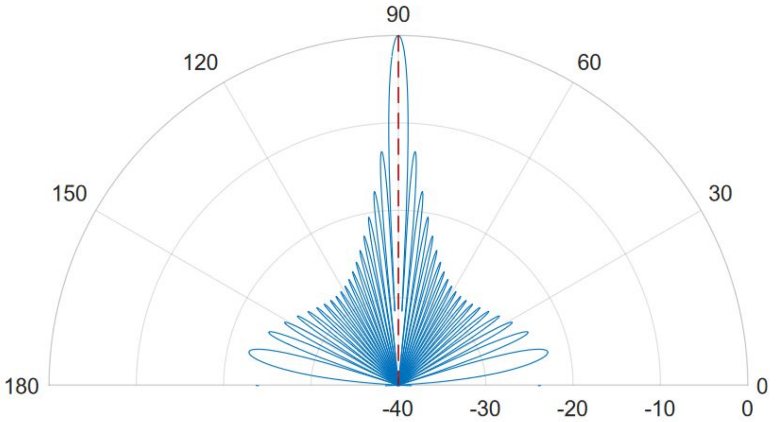
Delay-and-Sum Beamforming

The filter response is computed using the following expression:

$$|W(\psi,\theta)| = \left| \frac{\sin\left(N\pi \frac{fd(\cos\psi - \cos\theta)}{c}\right)}{N\sin\left(\pi \frac{\mathrm{fd}(\cos\psi - \cos\theta)}{c}\right)} \right|$$

Ψ: Incident angle of signal.
θ: “look-direction” - direction of sensitivity.

N: number of sensors.
d: distance between adjacent sensors.
c: signal propagation speed.
f: signal frequency.

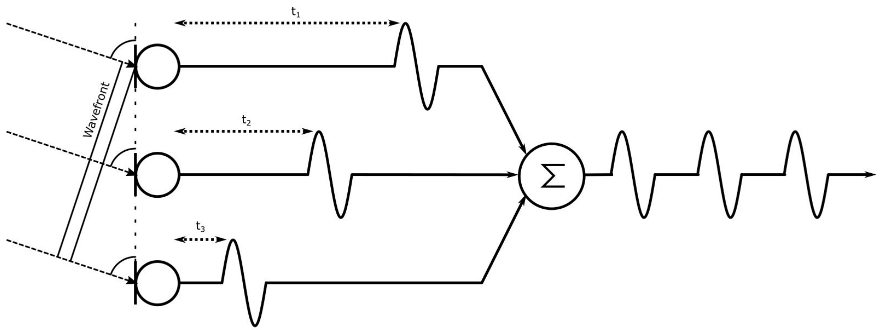


Theoretical beampattern modelled in MATLAB.
24 sensors, 25 kHz source signal

Delay-and-Sum Beamforming

In *linear, equidistant* sensor arrays, signals (waves) reach each adjacent sensor with a constant time-offset corresponding to the incident angle.

Time-offsets correspond to a *phase-offset*, leading to interference after summation.



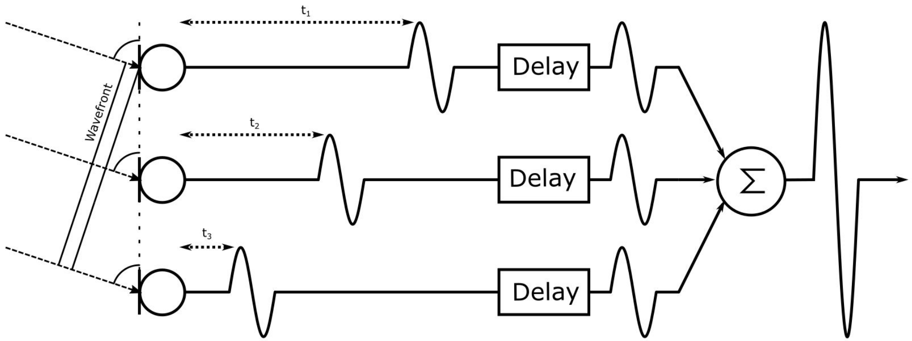
Summation without delay

Delay-and-Sum Beamforming

By delaying these signals by some amount, this phase-offset can be compensated for.

A *maximum output* value occurs when the signals are delayed such that they are all in phase.

A particular set of delay values that maximizes the output thus corresponds to the incident angle of the signal.



Summation after delay, showing constructive interference

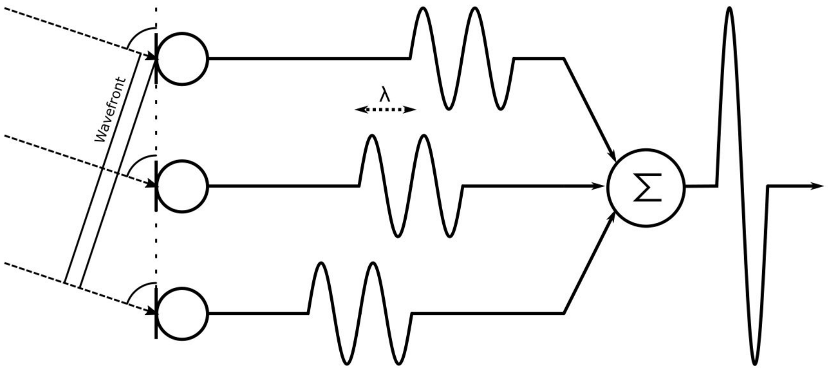
Delay-and-Sum Beamforming

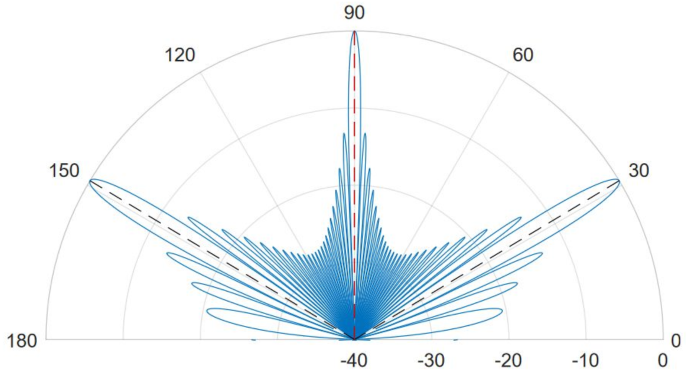
A fundamental limit of this technique comes in the form of *spatial aliasing* - if adjacent signals are offset by a multiple of a wavelength, it will appear as though they are in-phase!

Analogous to the Nyquist criterion, in order to avoid spatial aliasing, the array must satisfy

$$d \leq \frac{\lambda}{2} = \frac{c}{2f}$$

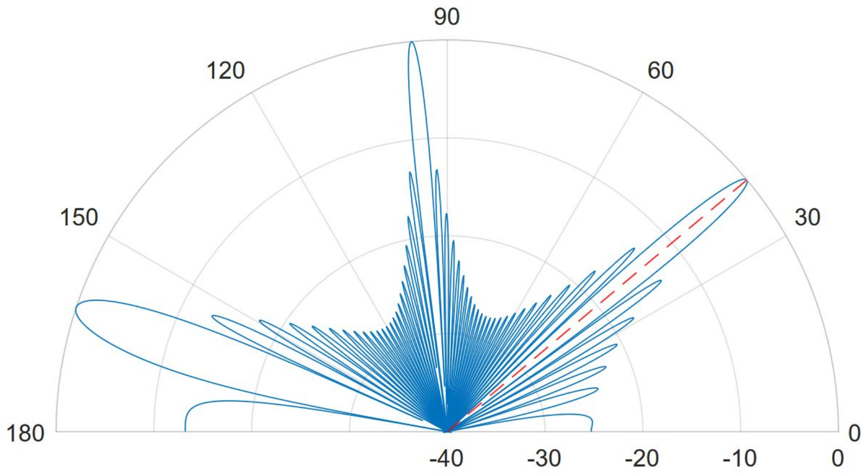
d: distance between adjacent sensors.
λ: signal wavelength.
c: signal propagation speed.
f: signal frequency.





Delay-and-Sum Beamforming

Since delays in DAS phase-shift by a constant, relative amount, aliasing cannot be eliminated through selection of delay values.



Delay Values

Delays (expressed as time-offsets) are calculated using for each sensor (indexed by n [0, N-1]):

$$\tau = \frac{nd}{c} cos(\theta)$$

n: current sensor.
d: distance between adjacent sensors.
c: signal propagation speed.
θ: "Look-direction" - direction of sensitivity.

Since we are working with a digital implementation, and thus discrete time units, this calculation must be adapted.

$$\tau_c = \lceil f_c \frac{nd}{c} cos(\theta) \rceil$$

fc: frequency of discrete time unit.

This gives the delay in terms of clock cycles.

Energy Values

The output of DAS is a summed signal, with a magnitude corresponding to the relative phase of the constituent signals.

In order to find the maximum, and thus the source direction, the *signal energy* must be calculated.

For a discrete signal, it is given by the following expression:

$$E_d = \sum_{-\infty}^{\infty} |x[n]|^2$$

Note! Our range clearly cannot be infinite, and so any calculation is an approximation.

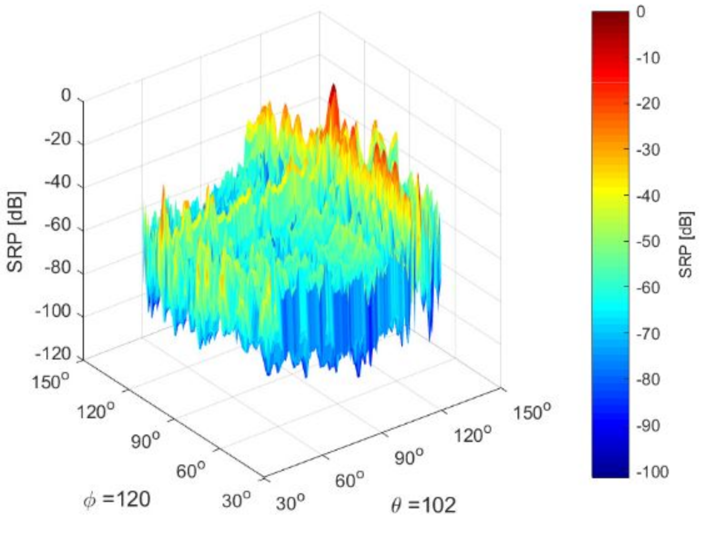
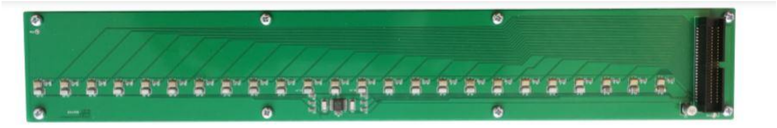


Fig. 4.8b)
Ultrasonic Source Localization with a Beamformer Implemented on an FPGA Using a High-density Microphone Array - Antonsson,Li

Hardware & Interfacing

Main Components

Xilinx Artix-7 FPGA AC701



Syntronic AB linear microphone array

Consists of 24 MEMS ultrasonic microphones, using PDM.

d = 11,43mm

GRLIB

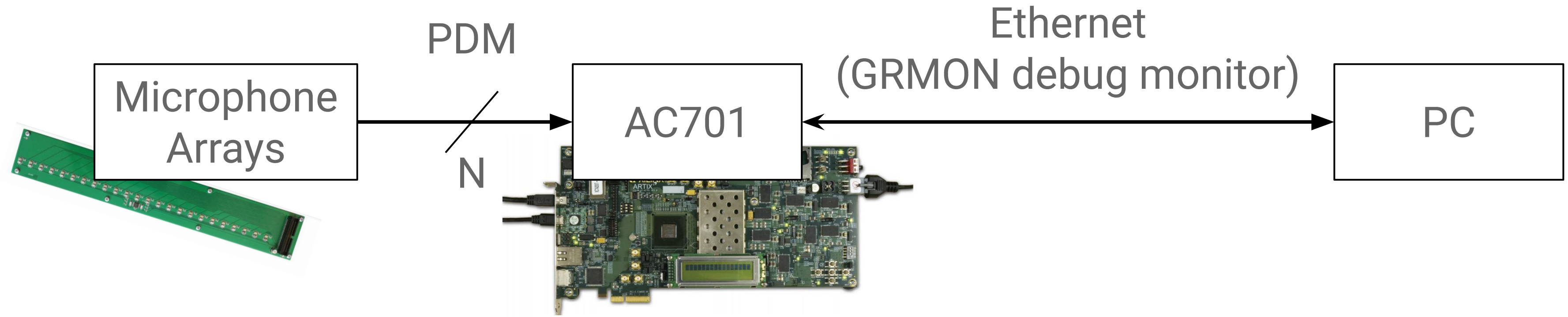
In order to effectively interface components, both internally and externally, support hardware must implemented on the FPGA.

This is provided in the Cobham-Gaisler Research Library (GRLIB), which contains many useful IPs.

Examples used in this project are:

- ethernet controllers
- AMBA AXI4 streaming-interface controllers
- RAM controllers
- CPU (LEON3) with debug monitor (GRMON)

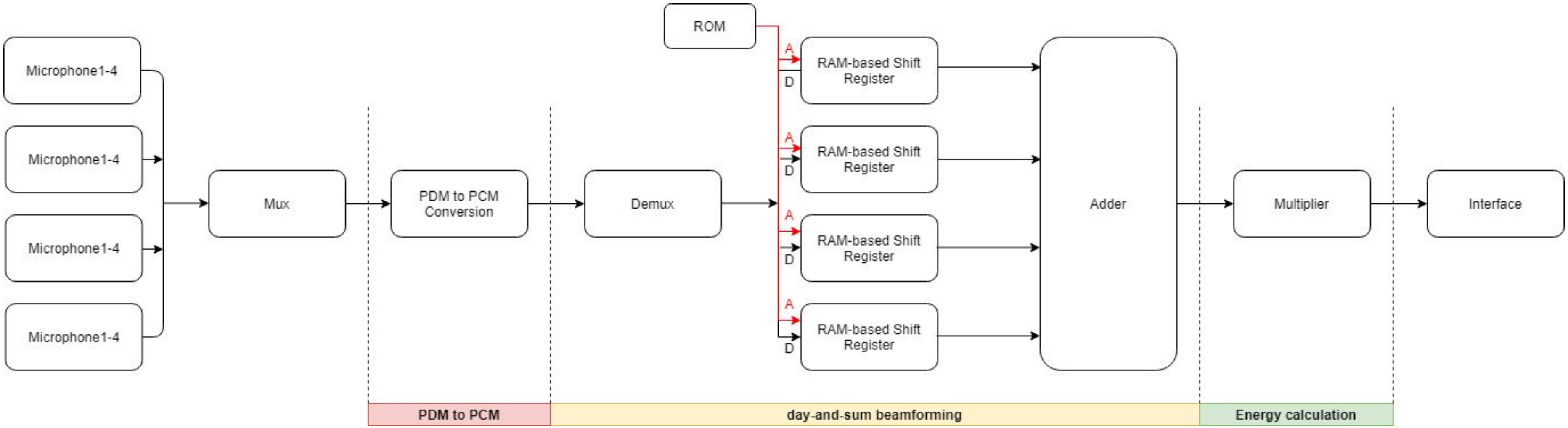
Data Flow



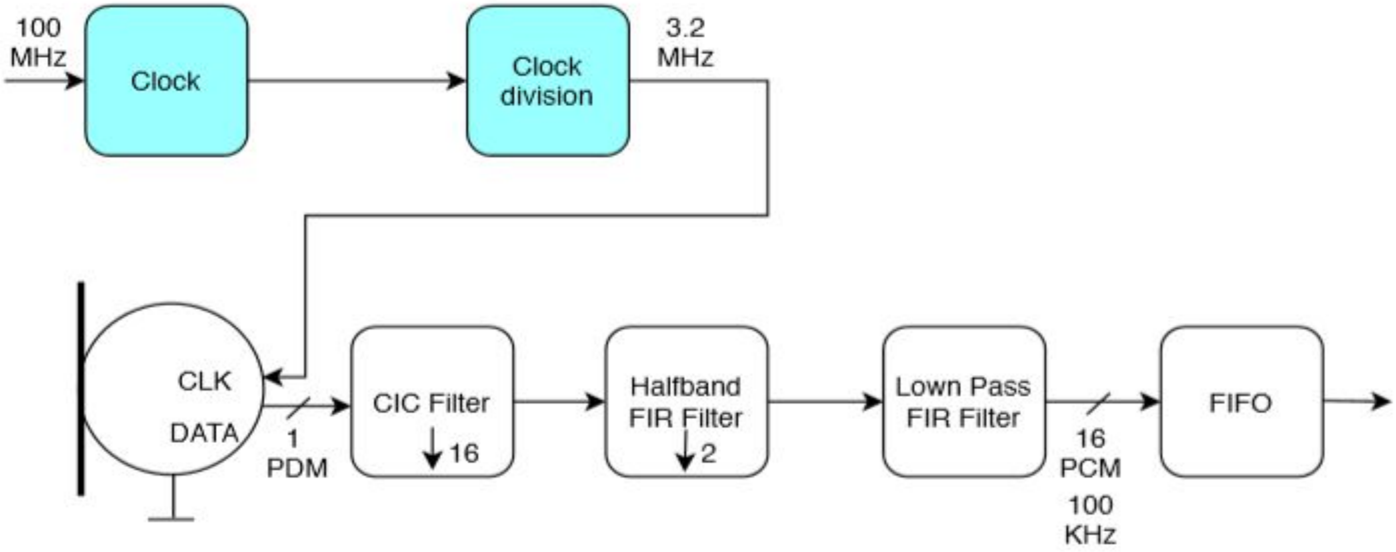
AMBA AXI4 used internally

Implementation

System Architecture

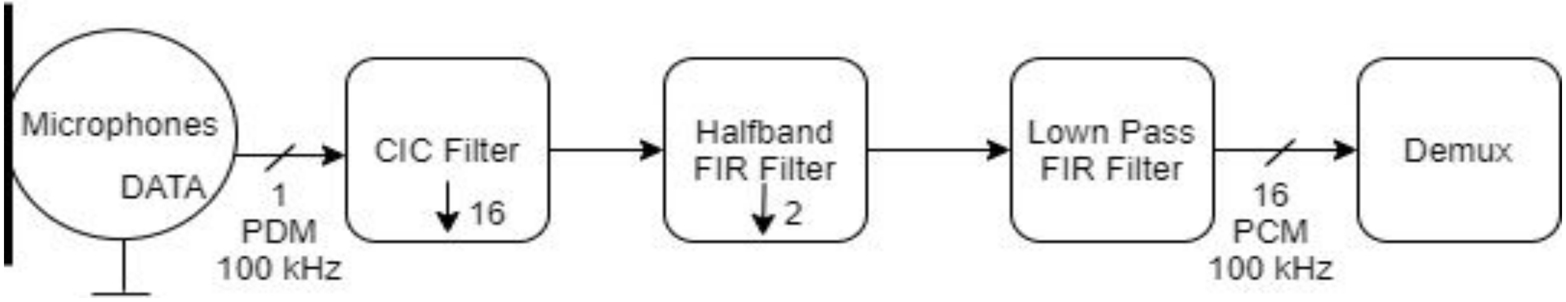


Clocking for Design



The PCM sampling rate in this design is 100 kHz and the PDM rate is 3.2 Mhz (100 Khz × 32) for each microphone. Note, microphones require >3.072 MHz for ultrasonic operation.

PDM to PCM Conversion



The conversion from PDM to PCM using one cascaded integrator-comb decimating filter(CIC filter), one half-band FIR filter, one low-pass FIR filter and one high-pass.

Filter coefficients are derived manually in MATLAB using the *FilterDesigner* tool and using filter-response parameters as inputs.

Mux and Demux

Mux:

Mux block is used to select one of 4 microphone input signals and forward the selected input to the output at the rising edge of the clock cycle.

Demux:

After the PDM data is successfully converted to PCM, we separate the data from the one channel form by using a demux.

The interleaved data is separated at the rising edge of the clock cycle again.

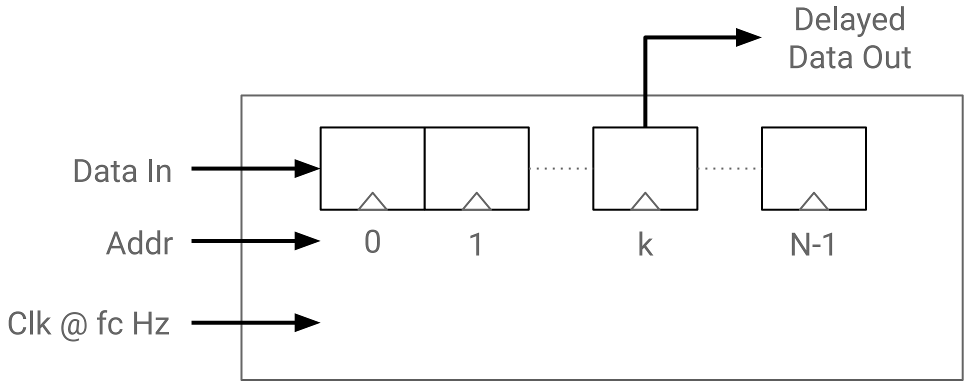
Delay Implementation

In this design, we beamform using statically defined look-directions (30°-150°).

$$\tau_c = \lceil f_c \frac{nd}{c} cos(\theta) \rceil$$

The PDM signals from the microphones are buffered in a fixed-length, addressable shift-register, and clocked (propagated) at the PDM frequency, fc.

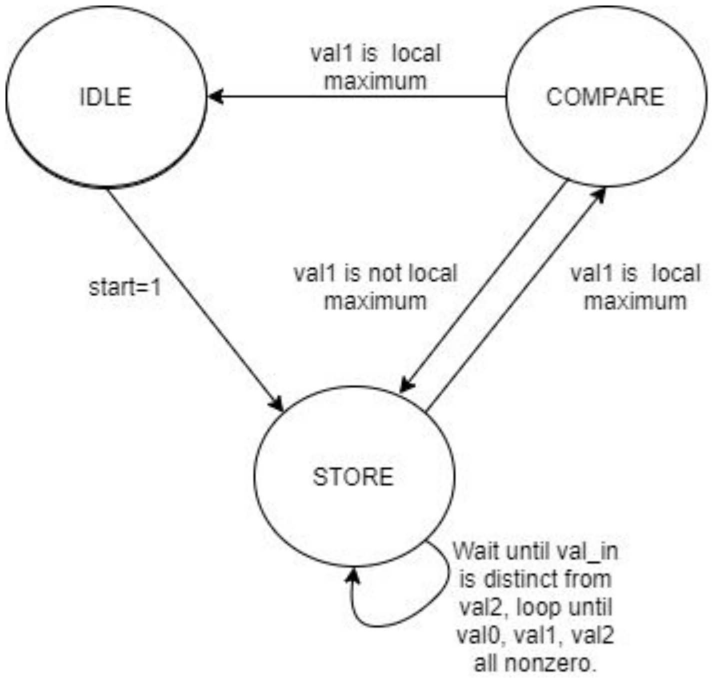
Delay values can thus be selected by addressing the shift-register with a certain index.



Energy Computation

In this design, the energy calculation is written with sum and multiplication together, which adds the output of delayed signal from shift register and squares the signal to get the energy.

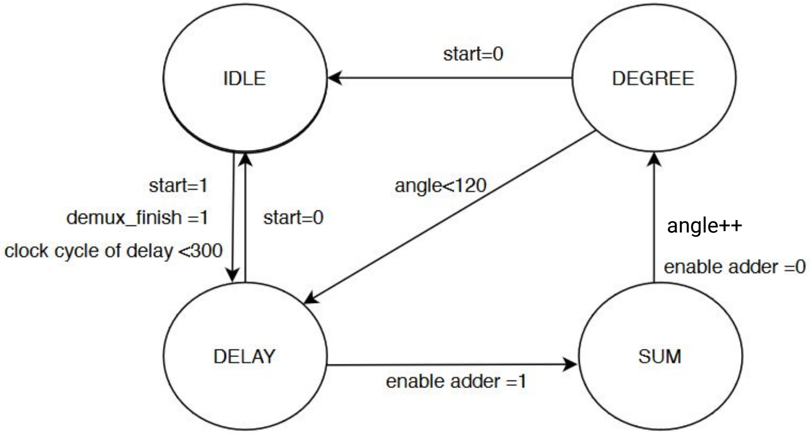
In order to extract the energy of the sound wave, we attempt to find the maxima of the squared wave. This is done by a simple algorithm to find local maxima. Points val0, val1, and val2, occurring successively, are compared. If val1 has the largest squared amplitude, it is a local maximum and is output through FIFO. Ten such values are extracted for each angle for the purpose of improving statistics.



Scanning Range of Angles

To scan a range of angles, an FSM (finite state machine) was implemented to “sweep” through the precomputed delay values.

In our system design, there are actually 120 sets of number of delay clock cycle corresponding to a scan between 30 degree to 150 degree. Thus, the counter needs to increment 119 times to finish the entire scan for the angles in a system



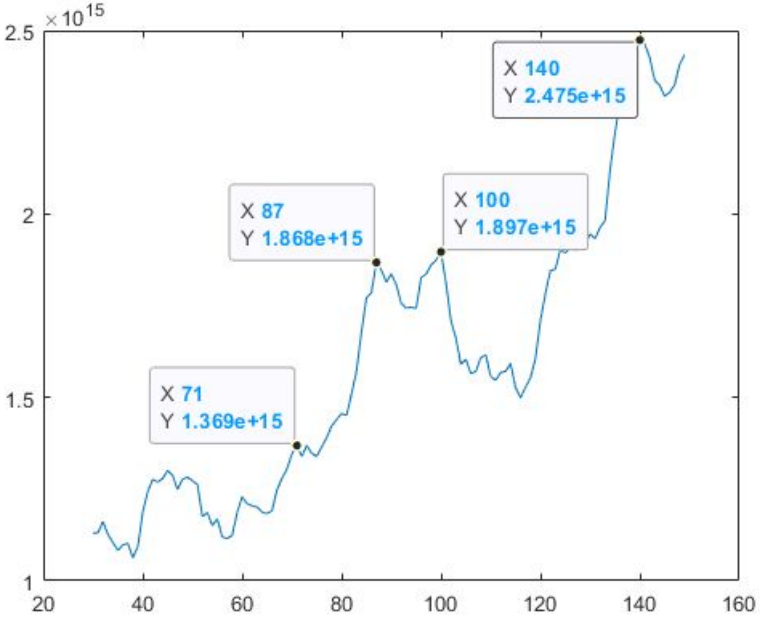
Results

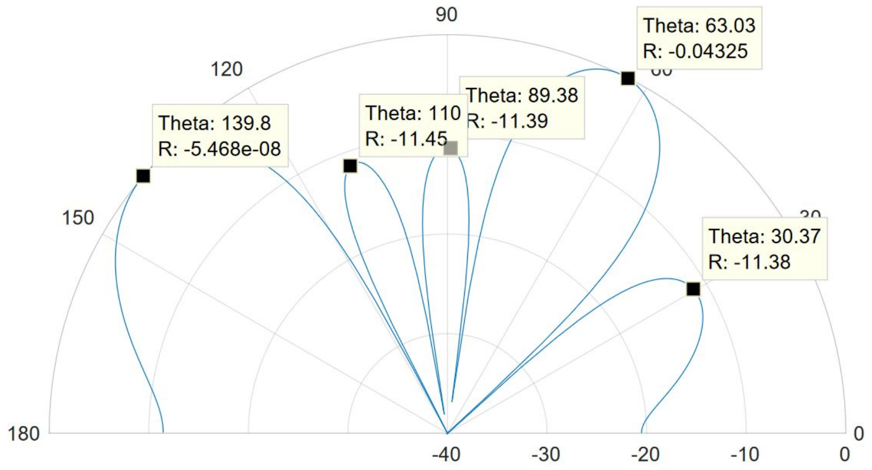
Experimental Data

The following slides show data obtained by the system, in comparison with theoretically expected values.

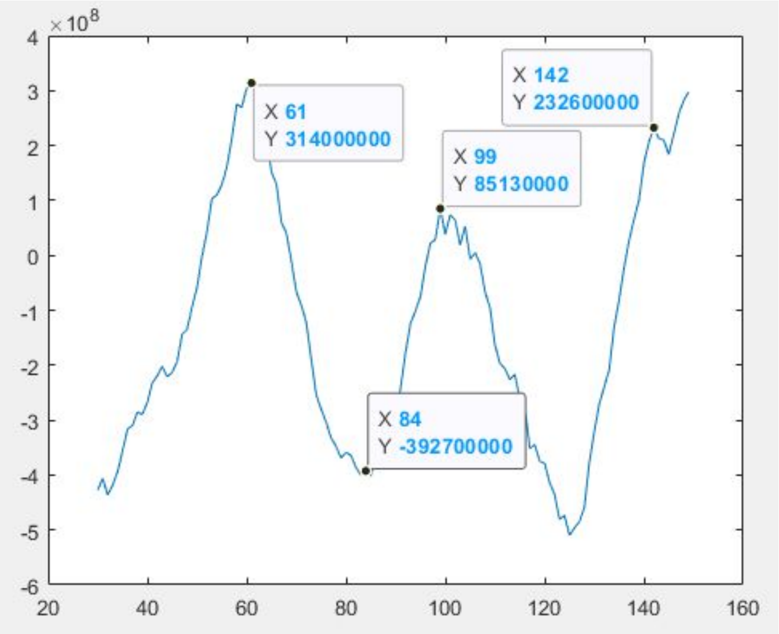
Energy values were written directly to memory, in order of increasing angle.

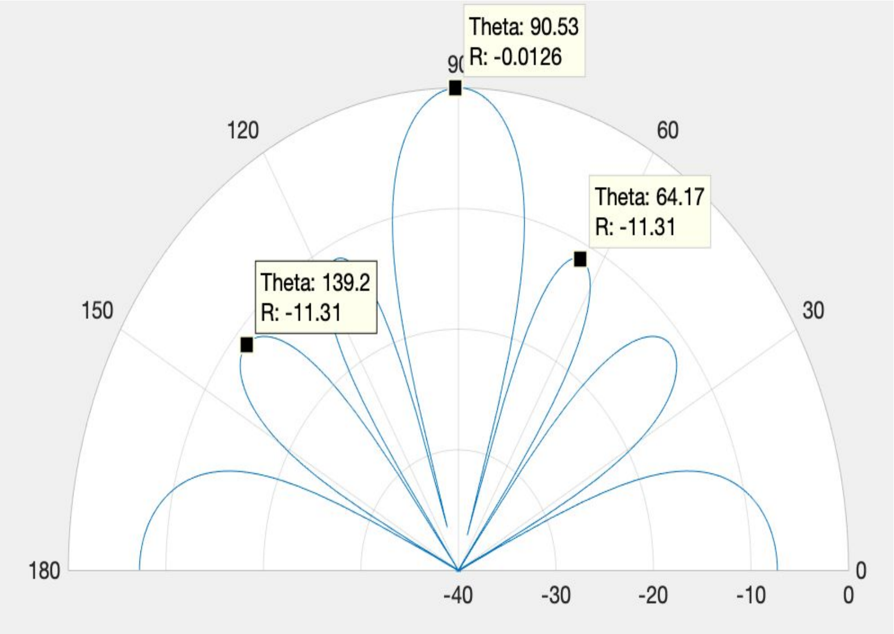
All experiments use 4 microphones and a source frequency of 25 kHz.





Comparison of energy data obtained (left) and theoretical beamforming calculation (right) at 140 degrees.





Comparison of energy data obtained (left) and theoretical beamforming calculation (right) at 90 degree.

Conclusions

Summary

- Beamforming results align fairly well with expected values, but suffer from unsolved issues, such as negative values..
- Not all design specifications were met
    - Lack of precision in results (>2°)
    - Real-time presentation of data.

Negative Values in Result

The system occasionally produces negative energy values, strongly implying wrapping due to insufficient data-widths.

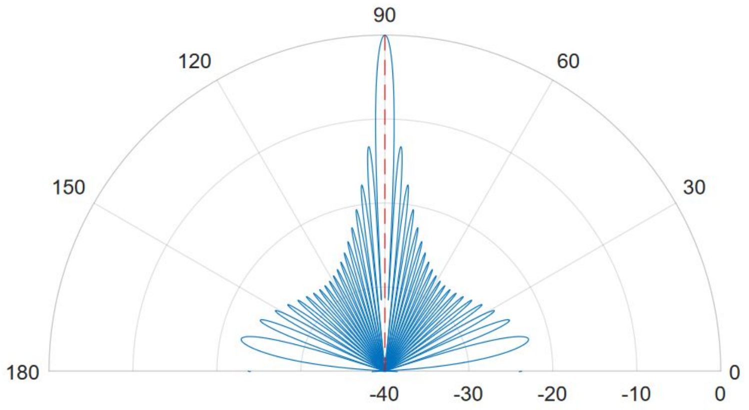
This can probably be solved by tracing data through the design and sizing data-widths after each computation such that overflow cannot occur.

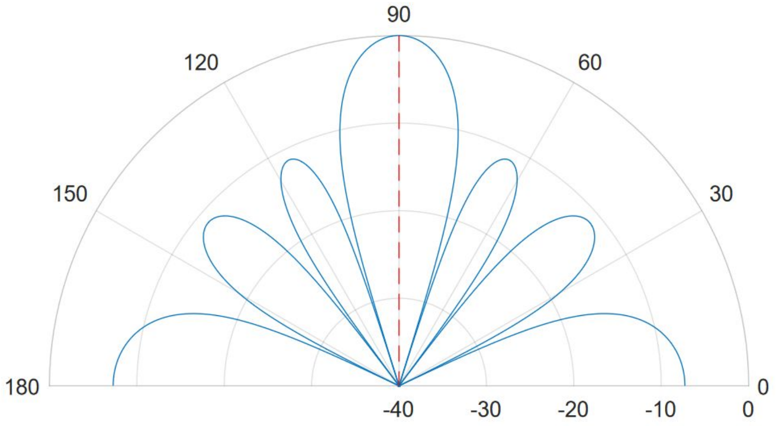
An alternative might be to use Integrated Logic Analyzer (ILA) to directly observe energy values and empirically determine the cause.

Unmet Specifications

- Lack of precision in results.
- Real-time presentation of data.

In order to increase beamforming precision, more sensors can be added to the system.
The effect on precision is clearly illustrated by the figures below, using 24 and 4 sensors, respectively.





Unmet Specifications

- Lack of precision in results.
- Real-time presentation of data.

Real-time presentation of data is challenging due to the GRMON interface.
In its current state, an operator must manually use GRMON to read memory contents.

As a partial solution, a script could be written using GRMON to periodically read incrementing blocks of data from the memory.

Future Work

Future Work

- Extending the system to use 24 sensors.
- Correcting internal data-widths so that overflow cannot occur.
- Using two sensor arrays to implement 2-dimensional beamforming.
- Implementation of continuous data transfer to PC when scanning, to enable real-time presentation of beamformer data.

Ending Notes

Unfortunate Circumstances

The ongoing pandemic* prompted the Swedish government to recommend social distancing and mandate remote education from LP4 onwards.

Since this project was a team effort, with shared physical hardware, the working methodology was adversely affected during the latter half of the project.

*https://en.wikipedia.org/wiki/COVID-19_pandemic

Thank you for listening!