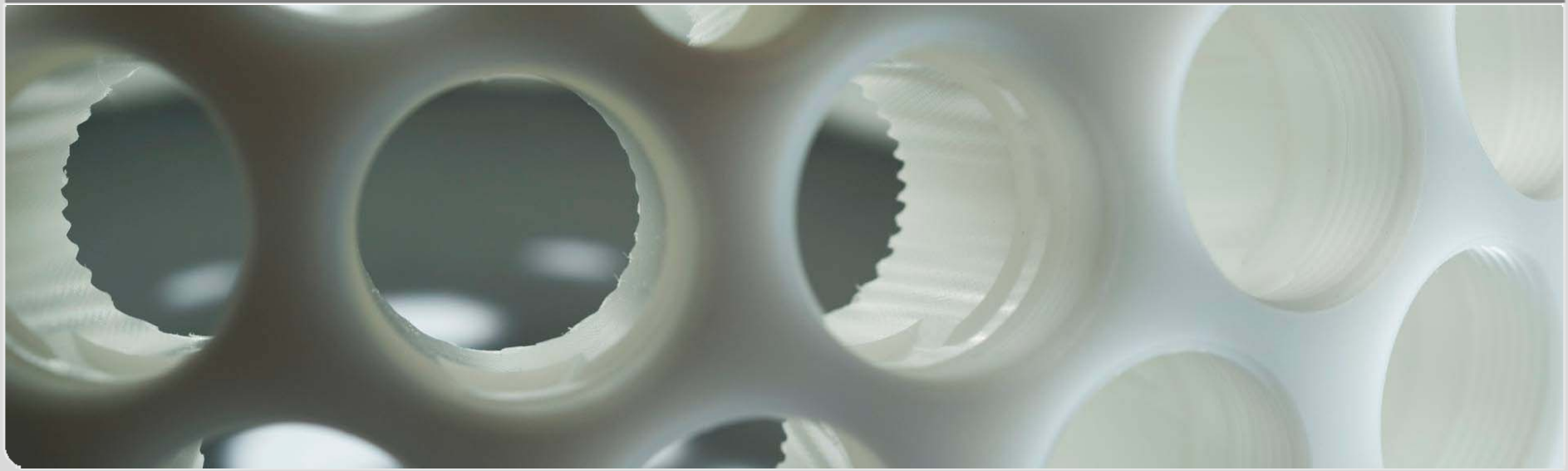
Optimal Virtual Aperture

Pain or pleasure.

M. Zapf, N.V. Ruiter

INSTITUTE OF DATA PROCESSING AND ELECTRONICS

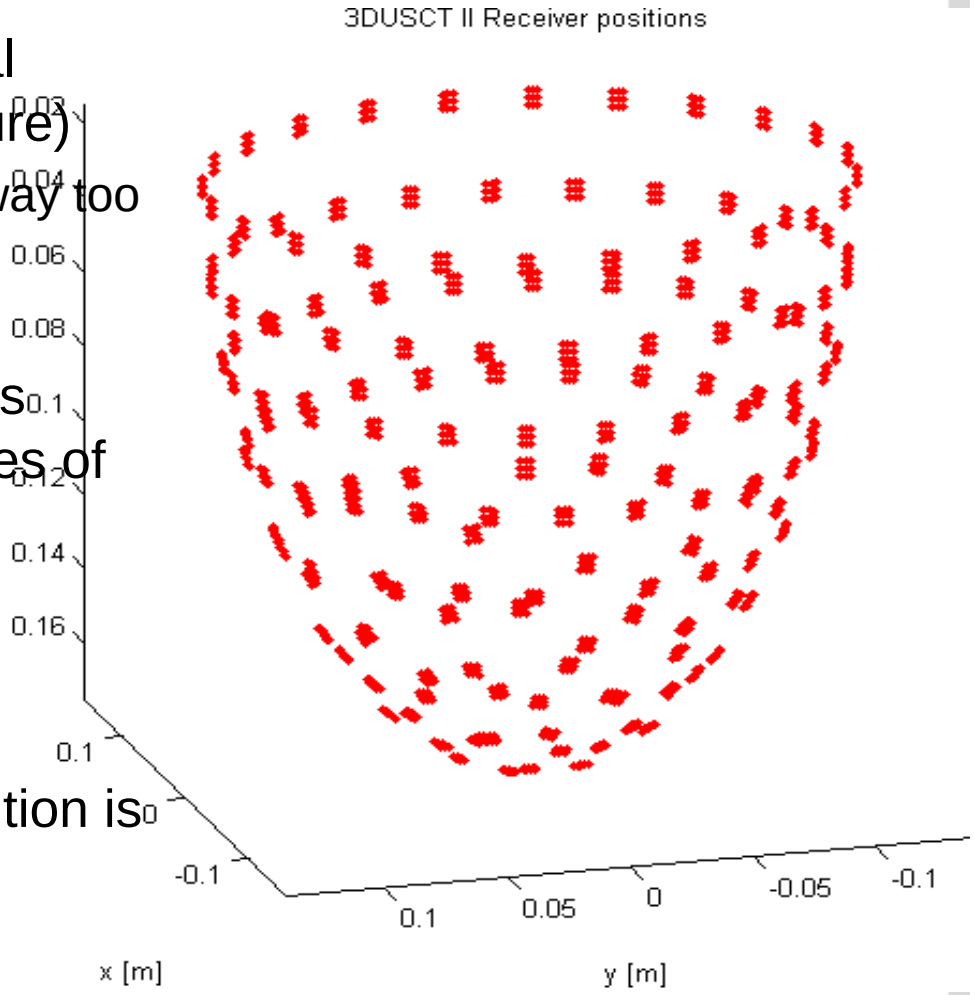


KIT – University of the State of Baden-Wuerttemberg and
National Laboratory of the Helmholtz Association

www.kit.edu

Motivation

- USCT has an highly sparse spatial sampling approach (sparse aperture)
  - (Or to name it directly, we have way too less transducers…)
- Therefore, to be able to reduce this sparsity of the aperture two degrees of freedom were added to the USCT
  - Rotation (+/- 20°)
  - Lift (0…0.03m)
- The moved aperture in a new position is called „virtual aperture“



Problem: What is a optimal Aperture?

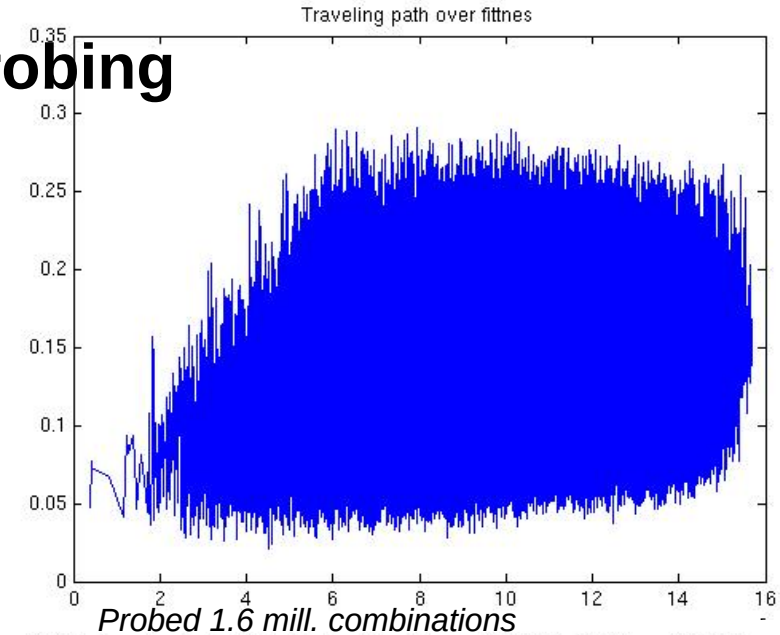
- „Optimality“ criteria not completly clear…
  - Maybe equidistanted? (homogeneous sampling of angles)
  - As non-periodic as possible? („Compressive sensing“-motivation)
  - Minimized travel-time? (reduce overall measurement time)

First approach: Evolutionary probing

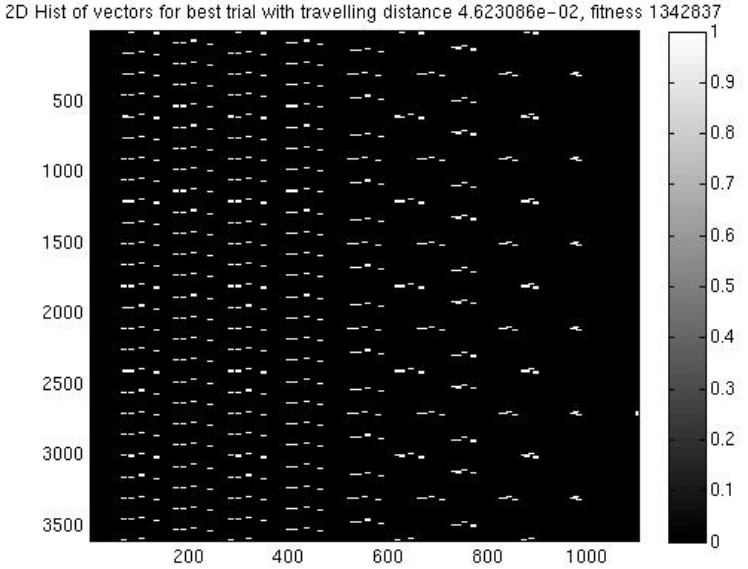
- Idea:
  - Maximizing the distance between all transducers
  - While minimizing the travel-way
- Problem: combinatoric explosion

- Evolutionary approach:
  - randomly „probing“ the space of possible 3-d virtual apertures
  - Adapting of good solutions

- Leads to satisfying results
  - still computational demanding (days)
  - not garantueed to find the „optimal“ solution



Probed 1.6 mill. combinations



2d projection: Selected best 4 virtual apertures

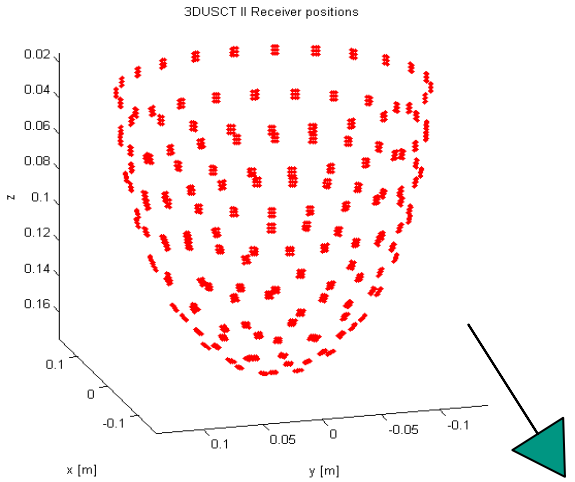
Second approach: Correlation formulation

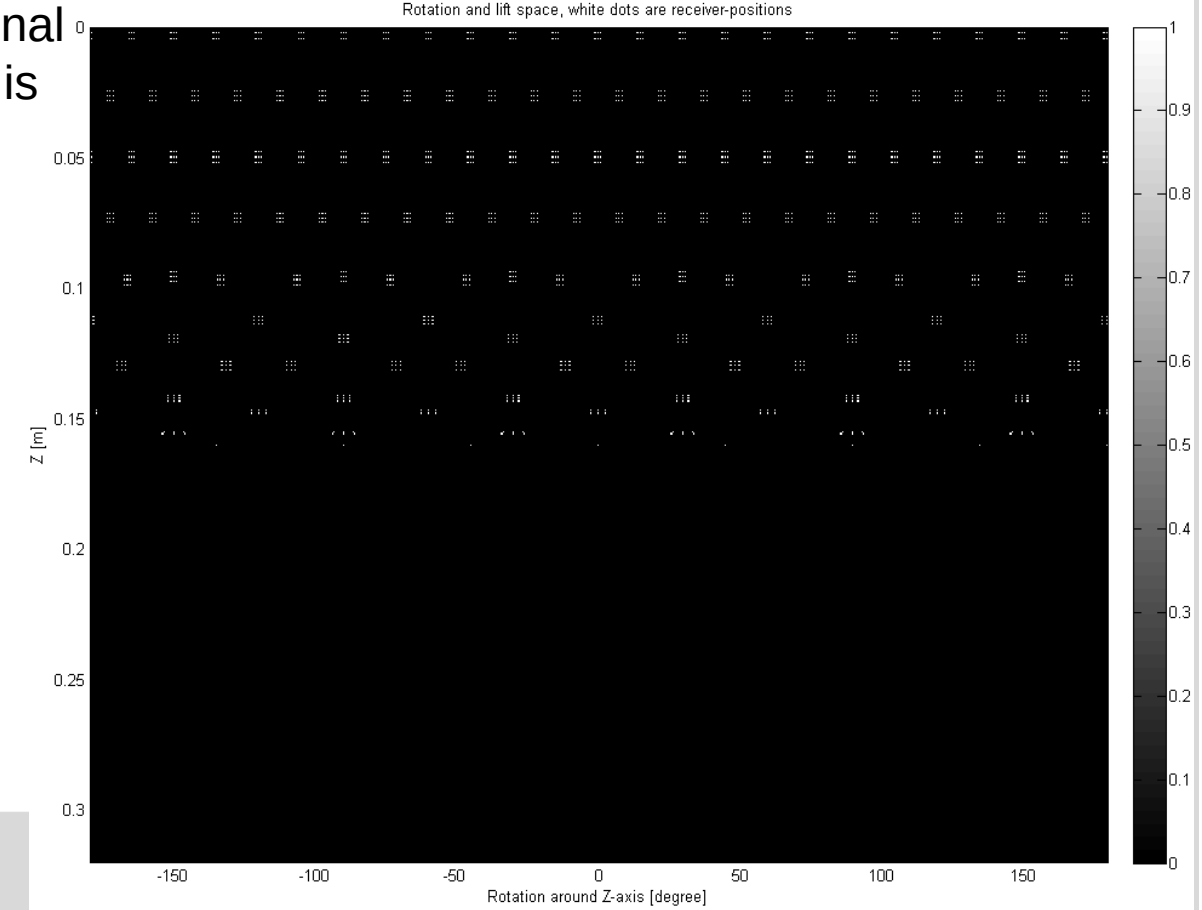
- Problem
  - For the study in Jena a new set of (bigger) virtual apertures was needed… fast!

- Idea:
  - A good virtual aperture is as much as possible „unsimilar“ to itself (or the worst VA is the aperture itself)
  - Well known problem in signal theory: autocorrelation gives some kind of self-similarity
  - A good virtual aperture has a minimal autocorellation in some domain
  - Our domain is defined by the two degrees of freedom roation and lift, therefore now a 2-d problem only

Transformation in 2D domain

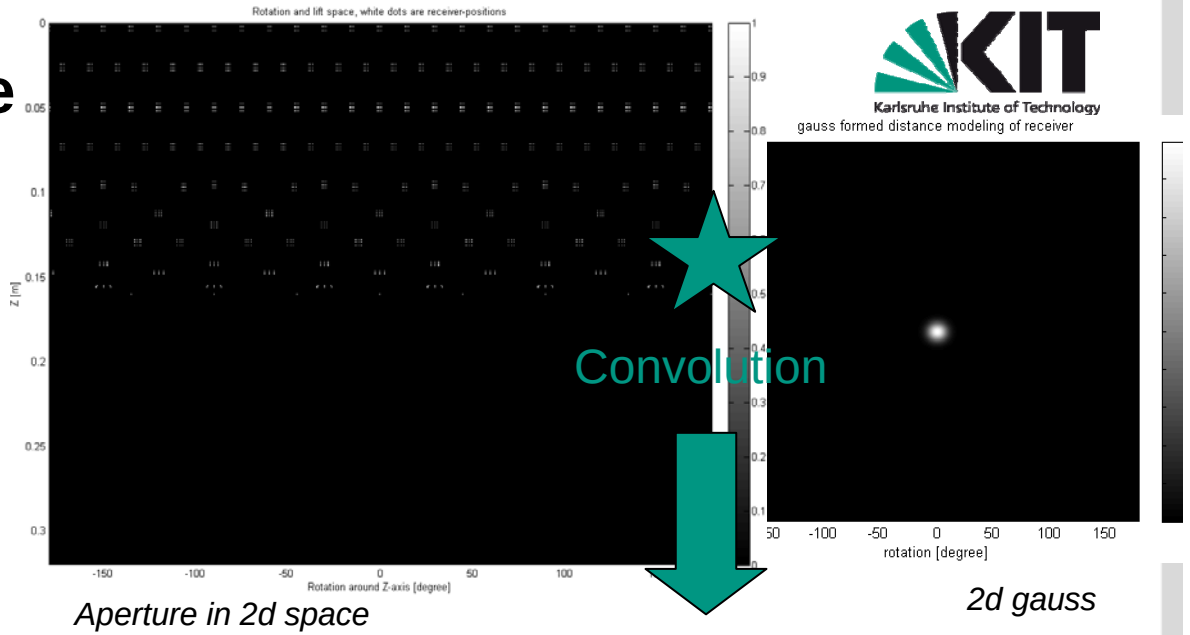
- Projection in 2D domain
  - Padding of Lift dimension to prevent „leaking“ (non-periodic dimension)
  - Non-padding of rotational dimension as this one is periodic





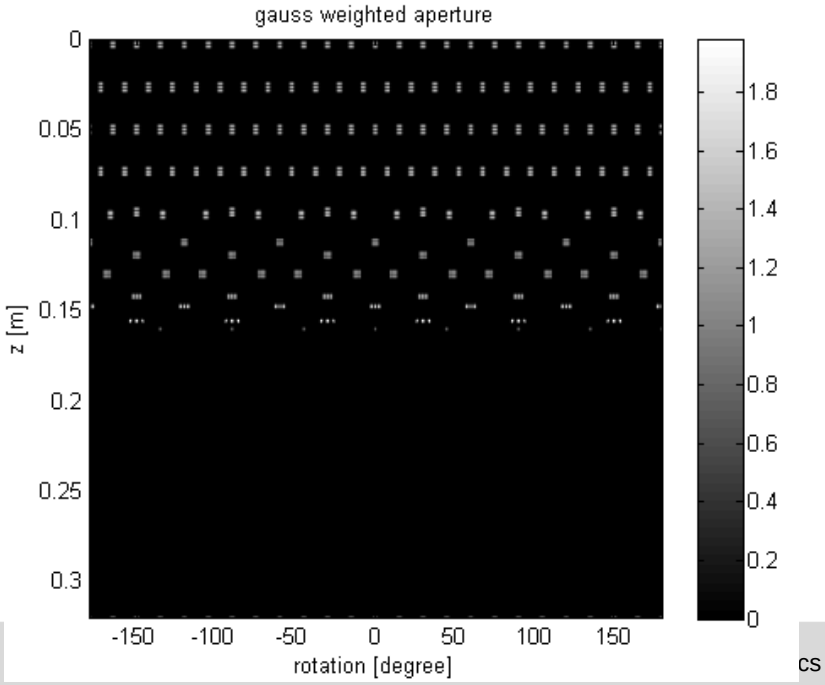
Modeling the distance

- Problem:
  - Aperture is represented as infinite small points
  - Correlation is indifferent for various non-equal positioned solutions
- Idea:
  - Give the aperture some spatial extend
  - Gauss weighting: more far away less „spatial influence“
  - Leads to „equidistanted“ solutions



Aperture in 2d space

2d gauss

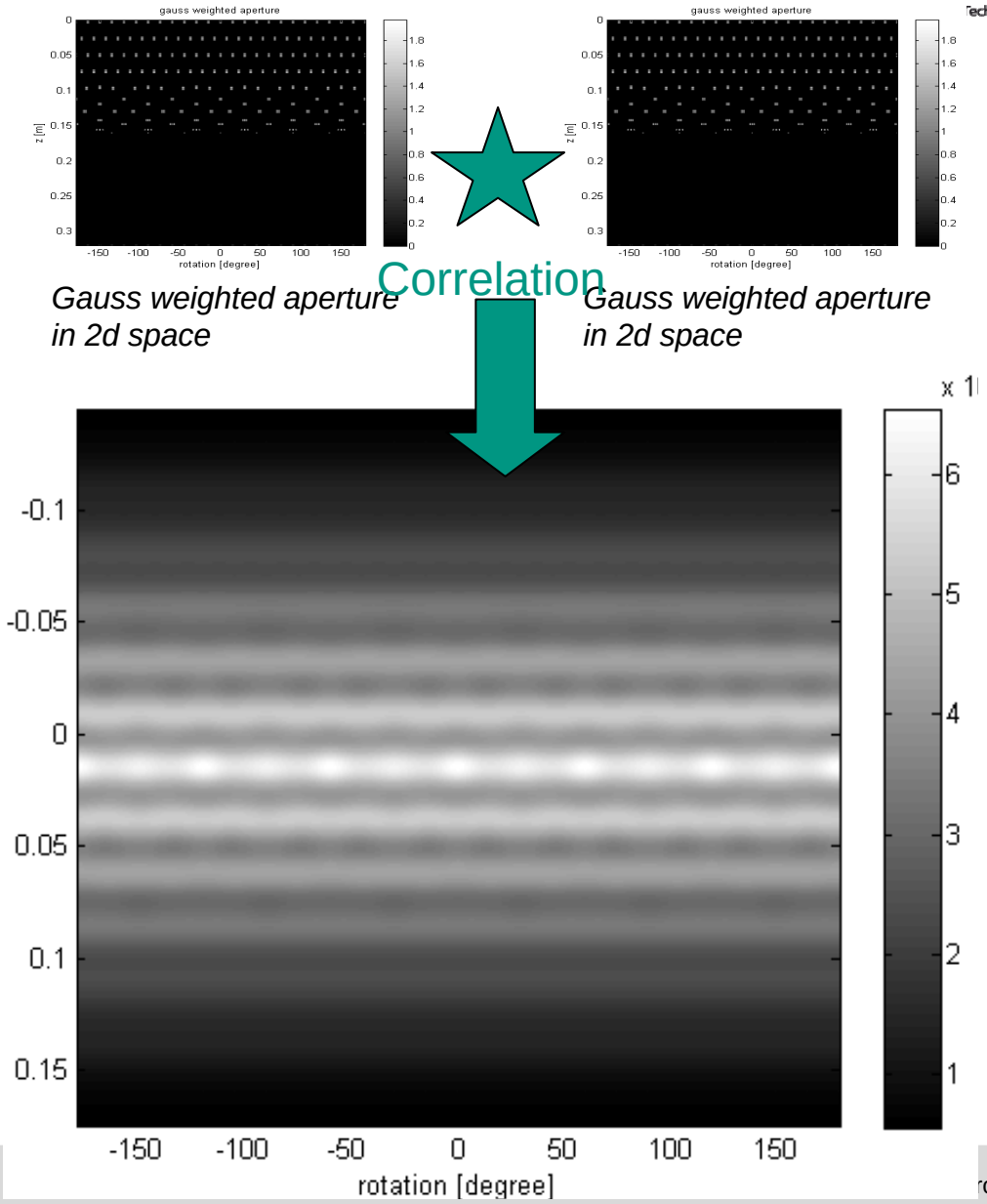


Gauss weighted aperture in 2d space

(auto-)Correlation

- Correlation of the apertures
- Minima are preferable virtual apertures
  - Lift dimension shows good properties
  - Rotation is pretty periodic

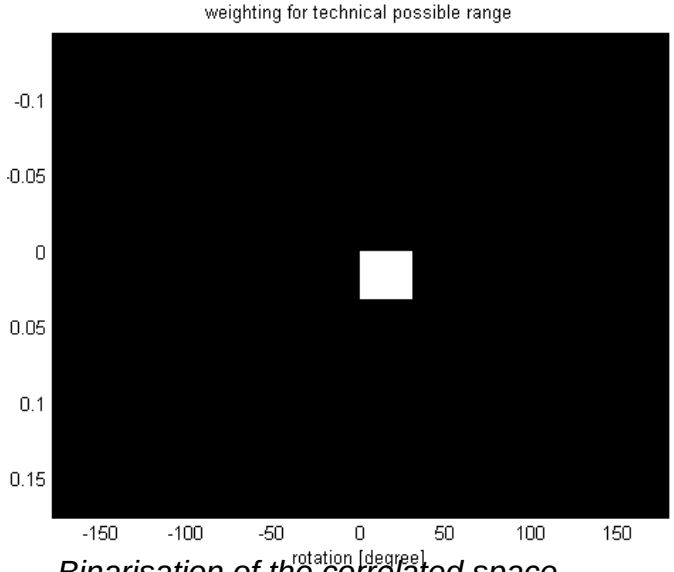
- Problem
  - Not all positions are possible or preferable



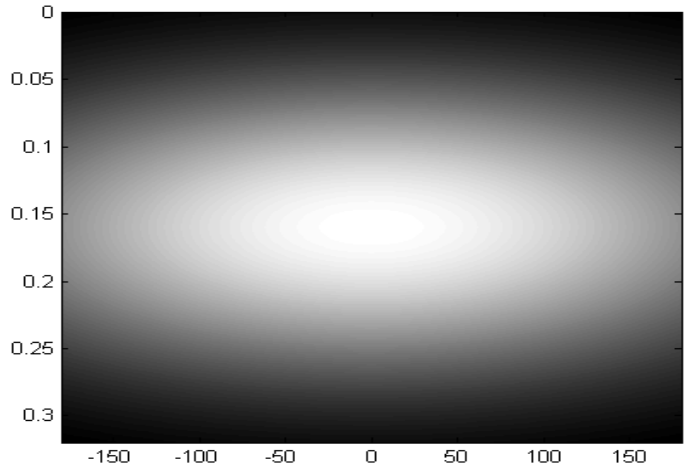
Refinement of the model

- Modeling of technical limitations
  - Weighting of the correlation with rectangular window

- Modeling of the movement costs
  - Weighting of the correlation with some a 2d gauss as distance function



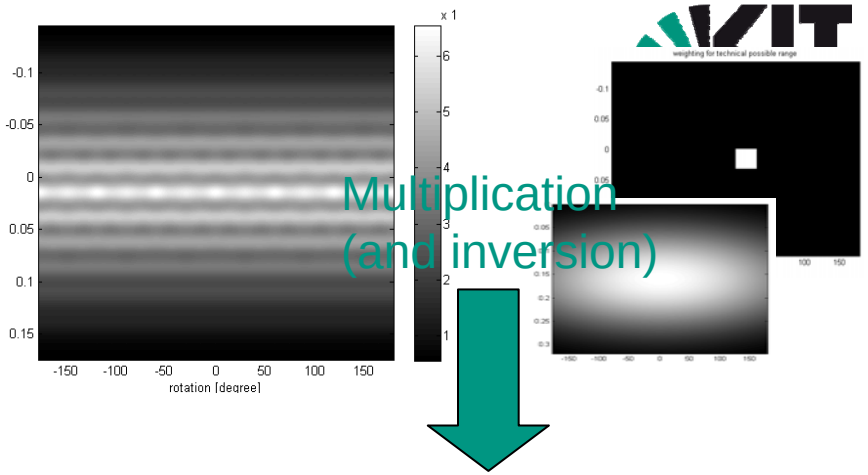
Binarisation of the correlated space

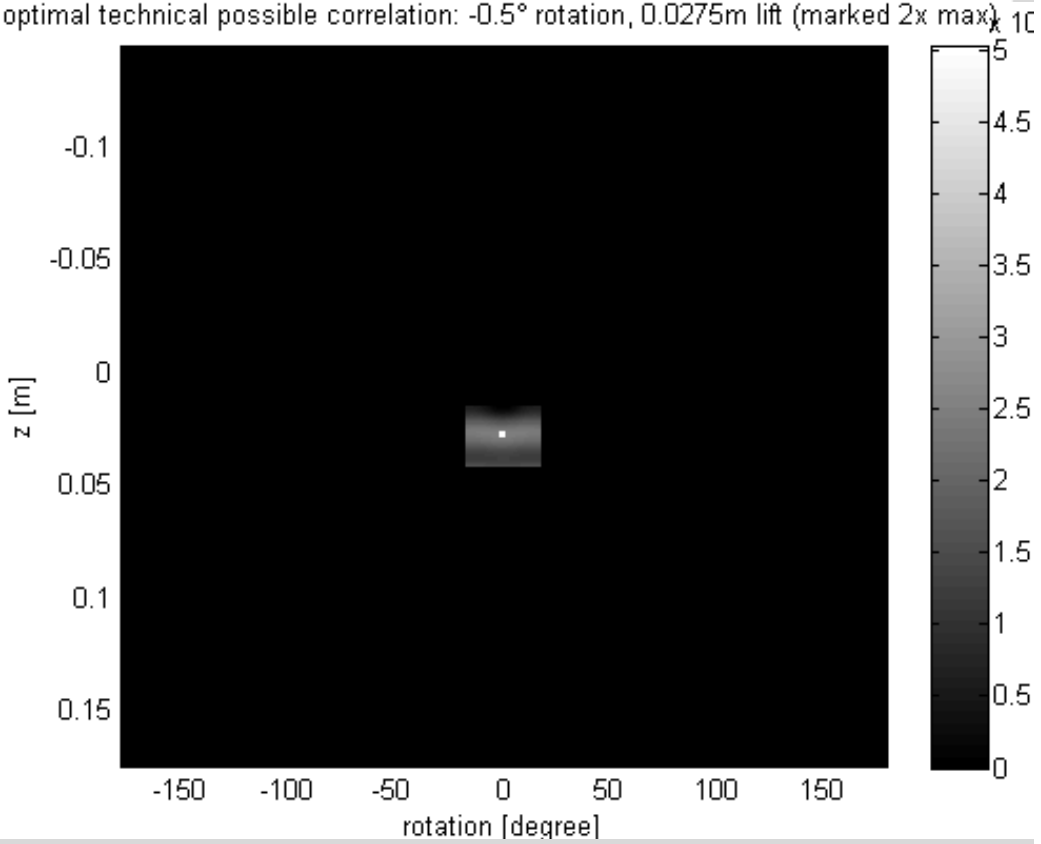


Gauss weighting around the actual aperture position (modeling of the movement cost)

Finding of Optimum

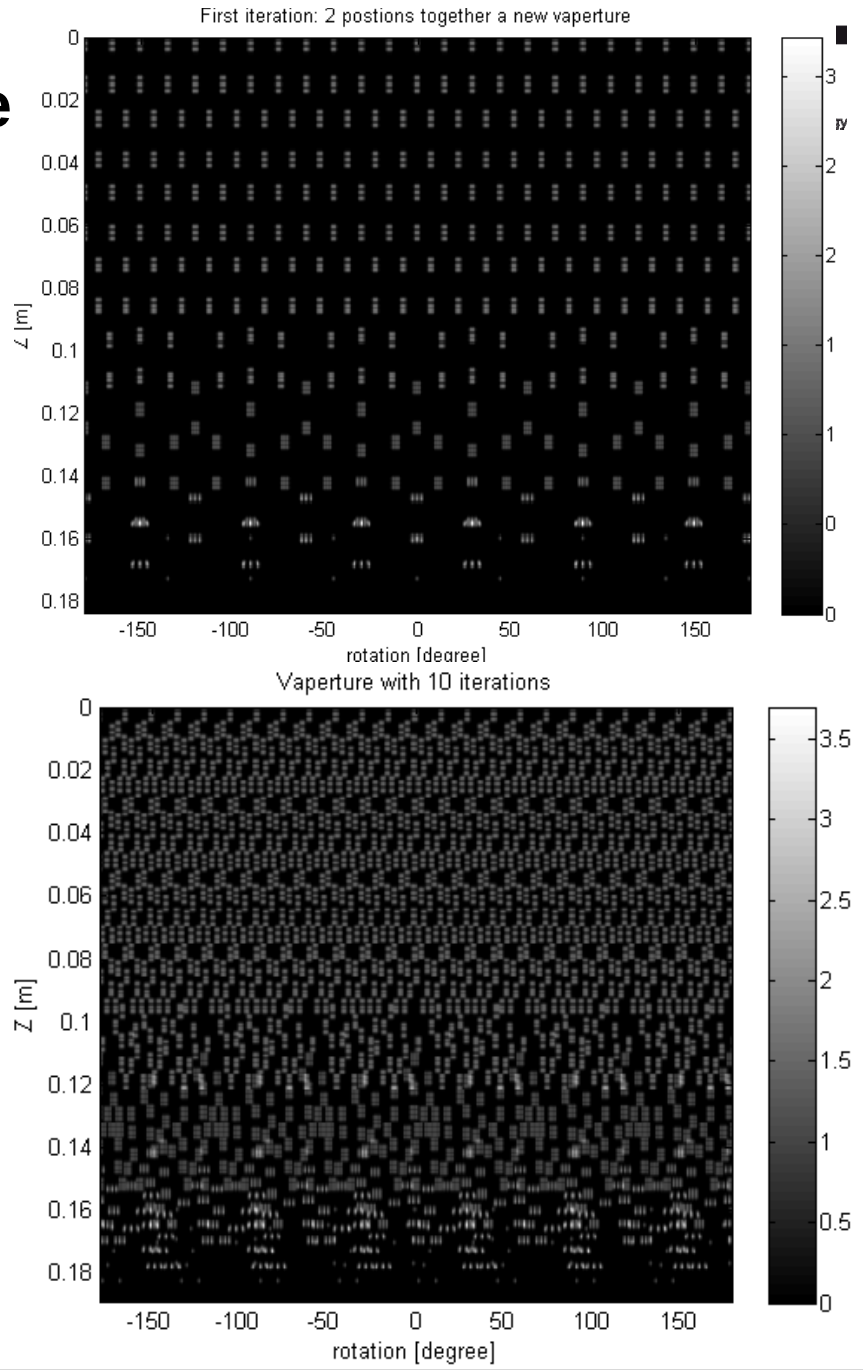
- Multiplication of weightings into correlation
- Inversion & search for maximum
  - In first step: 0.0125m lift and -0.5° rotation optimal





Creating of the new VAperture

- Adapting the aperture with the found lift and rotation and add this to the base aperture
- Continue this iterative process …

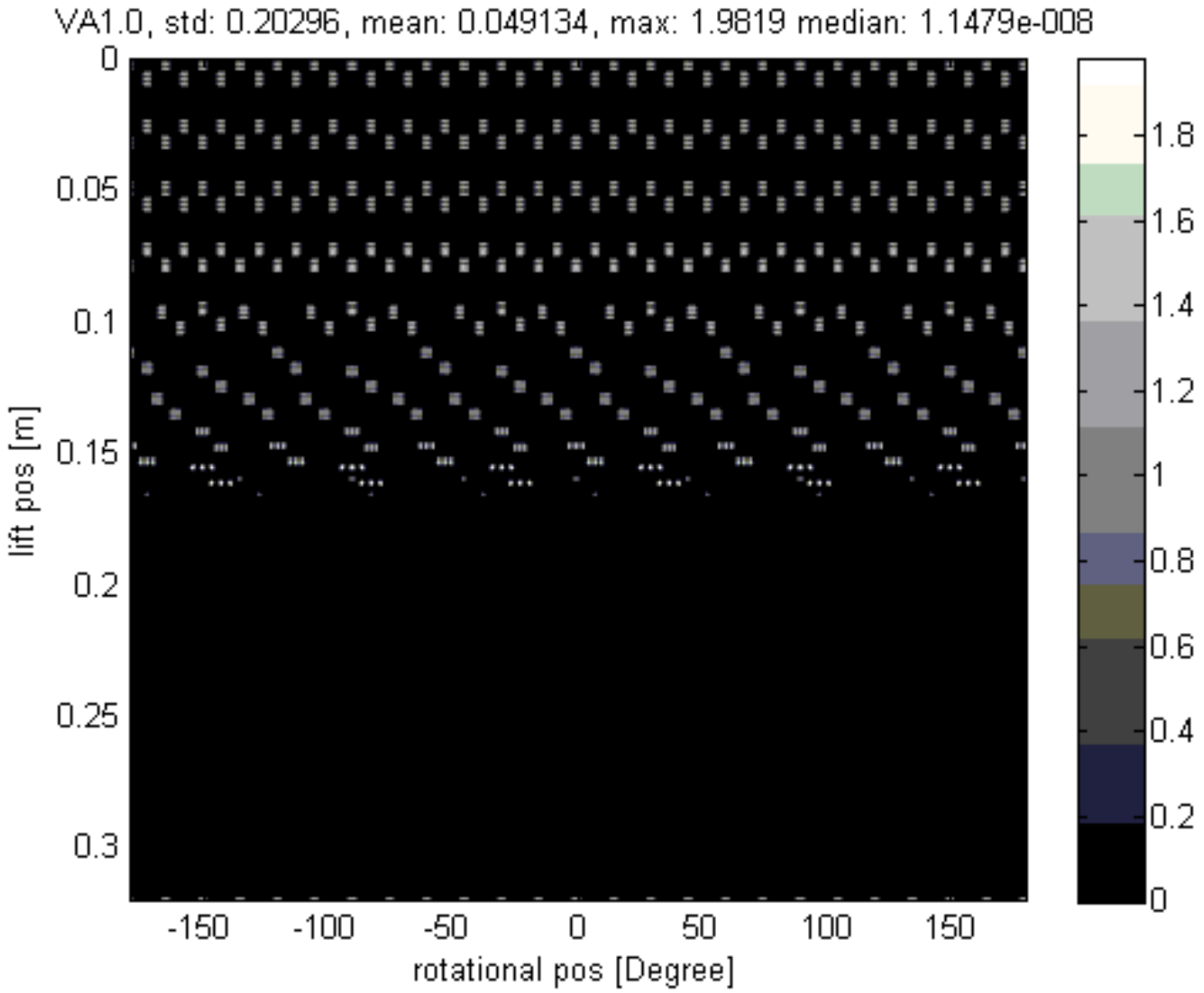


Results and conclusion

- Finds optimal (maximum mean equidistanted) virtual apertures
- Fast! (seconds per iteration)
- Modeling of technical limitations and travel cost possible

- Conclusions and Discussion
  - Rotation dimension offers especially in the first iterations limited gain -> aperture too periodic especially in the rotation dimension?
  - better performance of Lift dimension because auf more chaotic distribution?
  - Relation to compressive sensing and random distributed spatial sampling?
  - Resulting virtual aperture shows a reduction in periodicity
  - Autocorrelation a useful metric for evaluating apertures overall?

Thanks!

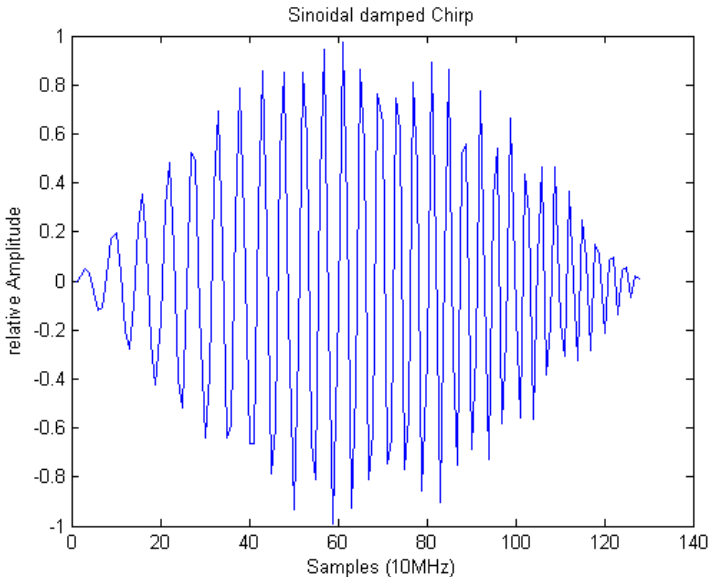

VA1.0, std: 0.20296, mean: 0.049134, max: 1.9819 median: 1.1479e-008
lift pos [m]
0
0.05
0.1
0.15
0.2
0.25
0.3
-150
-100
-50
0
50
100
150
rotational pos [Degree]
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0

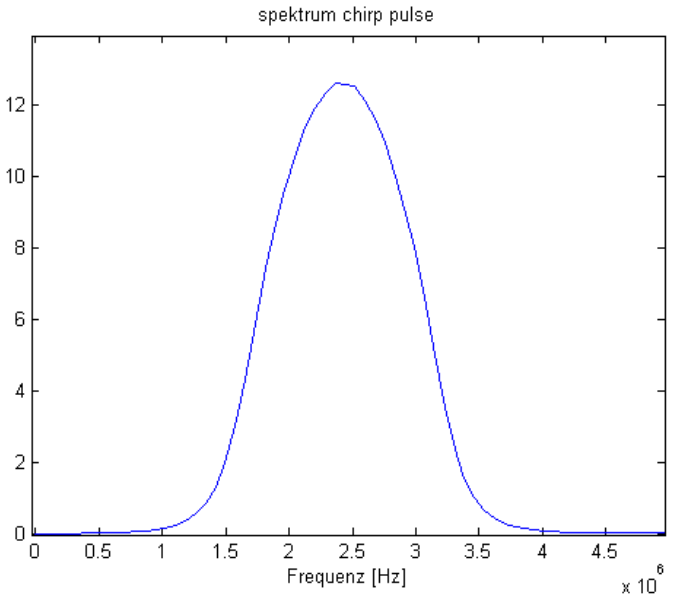
DAQ Constraints

- Transducer frequency sensitivity reaches from 1.3 MHz to 3.3 MHz (~95% drop-off BW)
- According to Shannon-Nyquist*:

$$f_{sampling} = 2 \times f_{\max}$$

- For DAQ 6.6 MHz would be fine…
  - ADCs 20MHz
  - FLT data storage to DDR in 10MHz
- 1/3 is not enough….

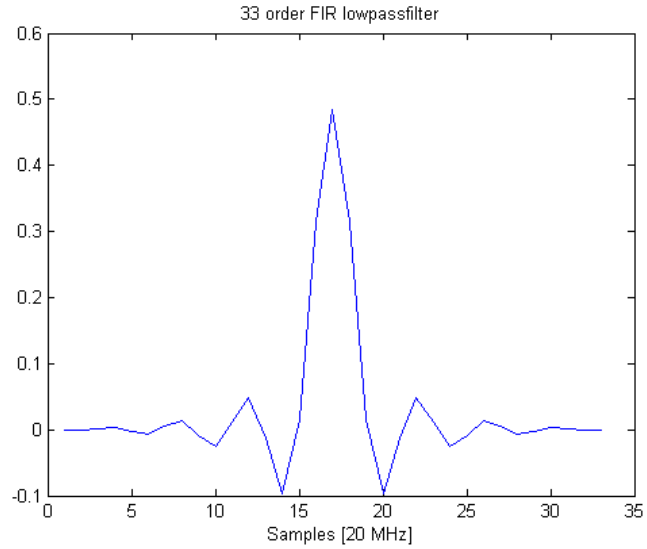


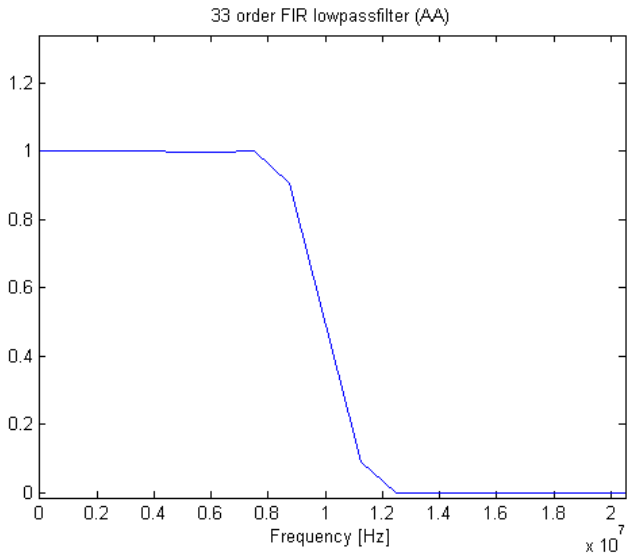


* if the lowest frequency is zero

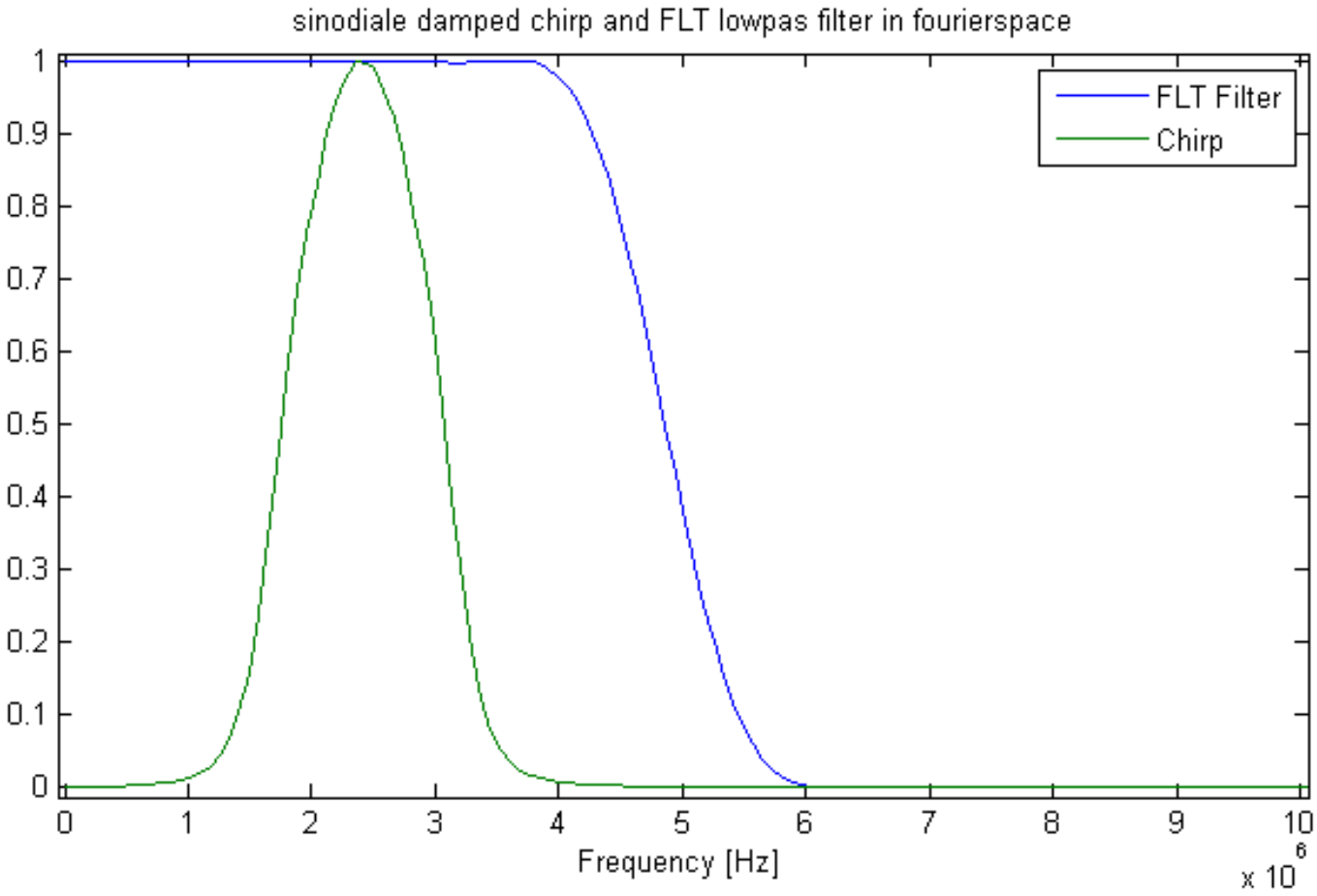
FLT DAQ Chain

- ADC 12Bit, 20MHz
- FLT
  - 32 order FIR antialiasing filter (lowpass)
  - Decimation by 2, basically throwing samples away
  - Averaging (max 256 times)
  - Storing as 16Bit, 10MHz in DDR





Wasting of „Fourier space“?



- Actually, only 1/3 of the sampled frequency bands are used!

Back to the basics: Nyquist II

- Shannon-Nyquist*:

$$f_{sampling} = 2 \times f_{\max} = 6{,}6MHz$$

* if the lowest frequency is zero!!!

- General Shannon-Nyquist:

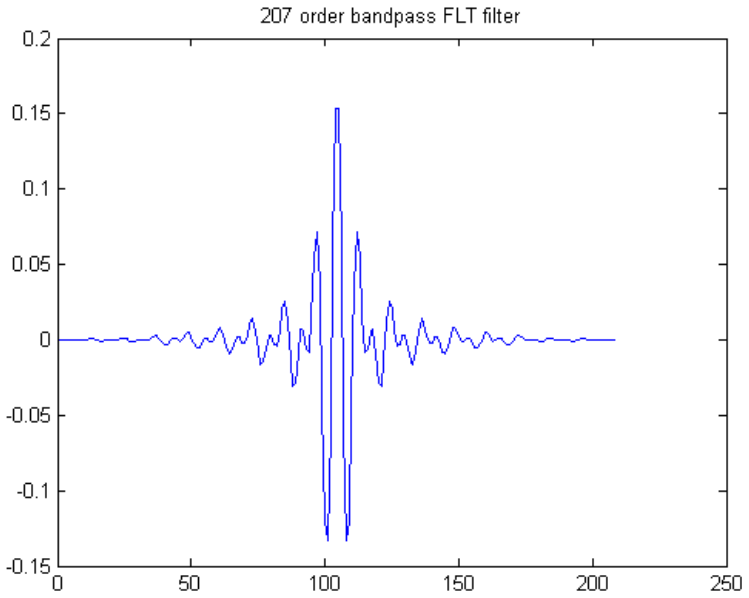
$$f_{sampling} = 2 \times (f_{\max} - f_{\min})$$

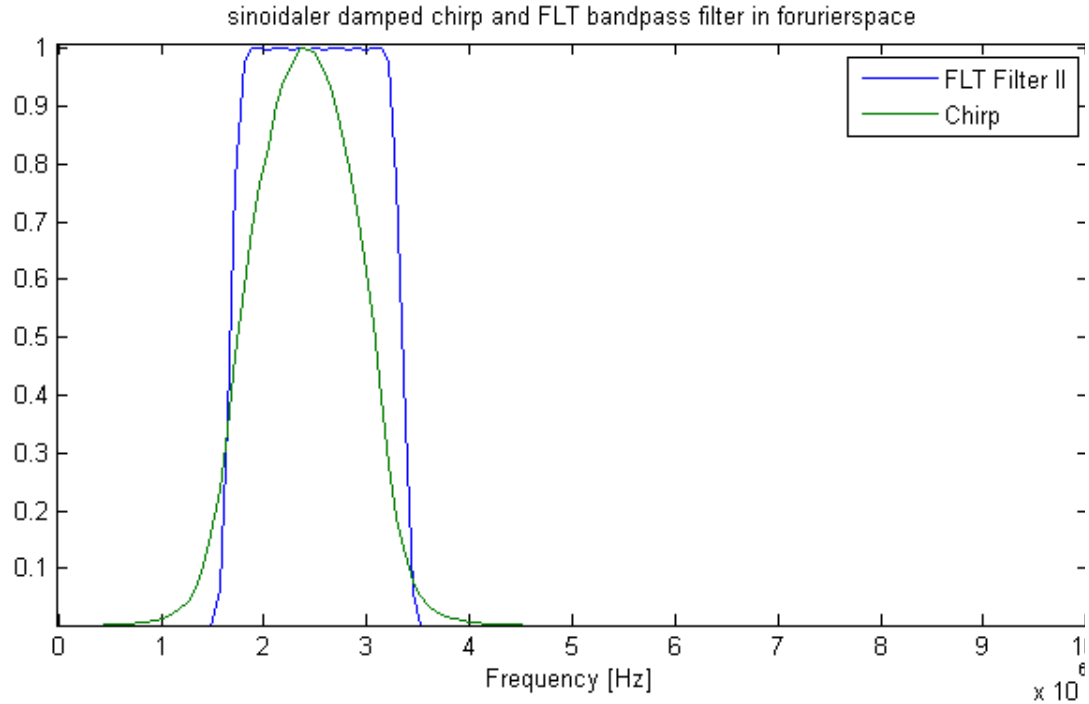
$$f_{sampling} = 2 \times (3.3MHz - 1.3MHz) = 4{,}0MHz$$

- How can this be exploited in the existing setup?

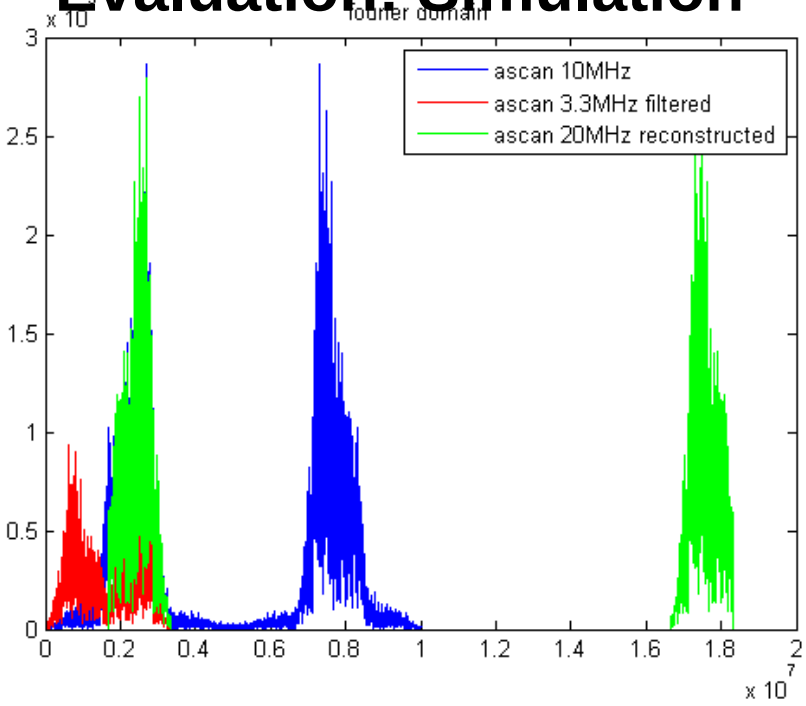
Idea: Bandpass subsampling

- Just exchange parameters: FLT filter coefficients and decimation step width!
- Roughly 1/3 of the fourierspace is used:
  - Instead of decimation 2, decimation 6
- Instead of a lowpass filter, a bandpass filter
  - „Reserves“ the lower band for the high-freqs which alias into the lower band (not seperateable anymore!)
- Additionally, increase the filter order from 32 to 207 (sharper edges)

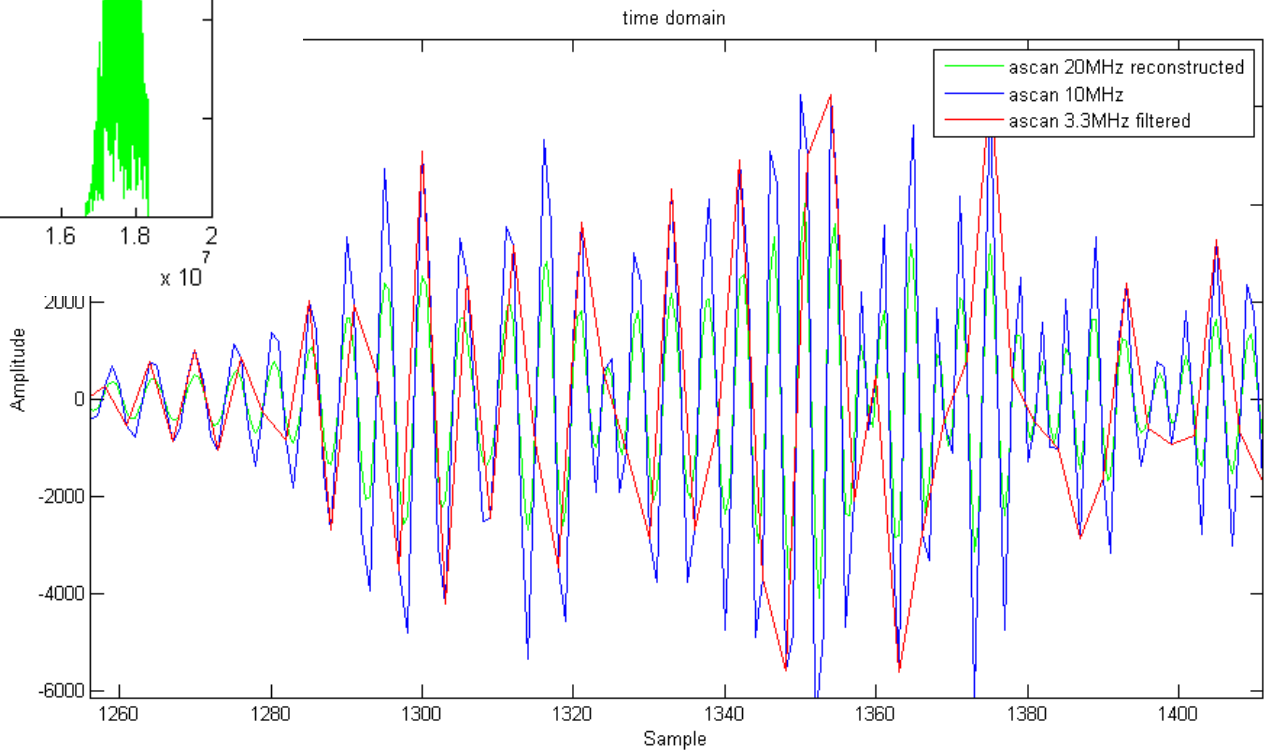




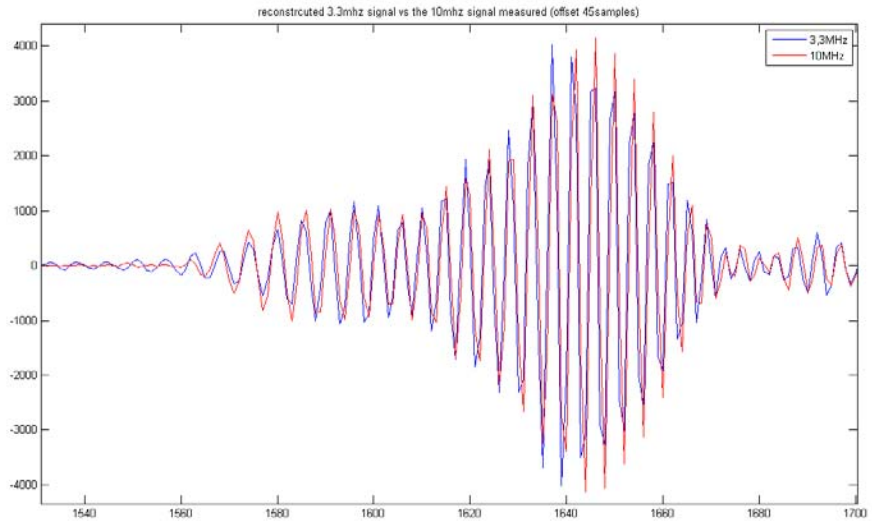
Evaluation: Simulation

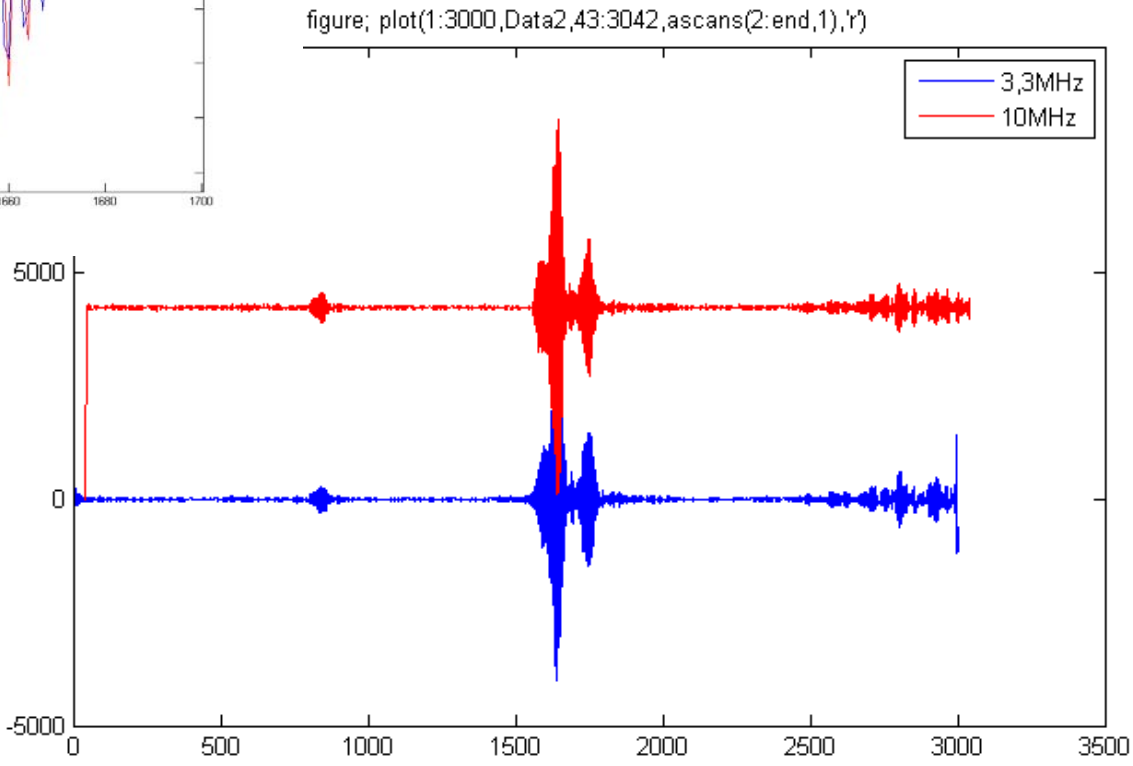


- Simulation with empty measurement signals (ascans)
  - Up-sampled to 20MHz
  - Applied second filter
  - decimated by 6 to 3.3MHz
  - Reconstucted to 20MHz



Evaluation: Real Measurements





- Two sequential empty measurements
  - first with 10MHz filter
  - Second with 3.3MHz filter
  - Afterward reconstruction in software

Result & Discussion

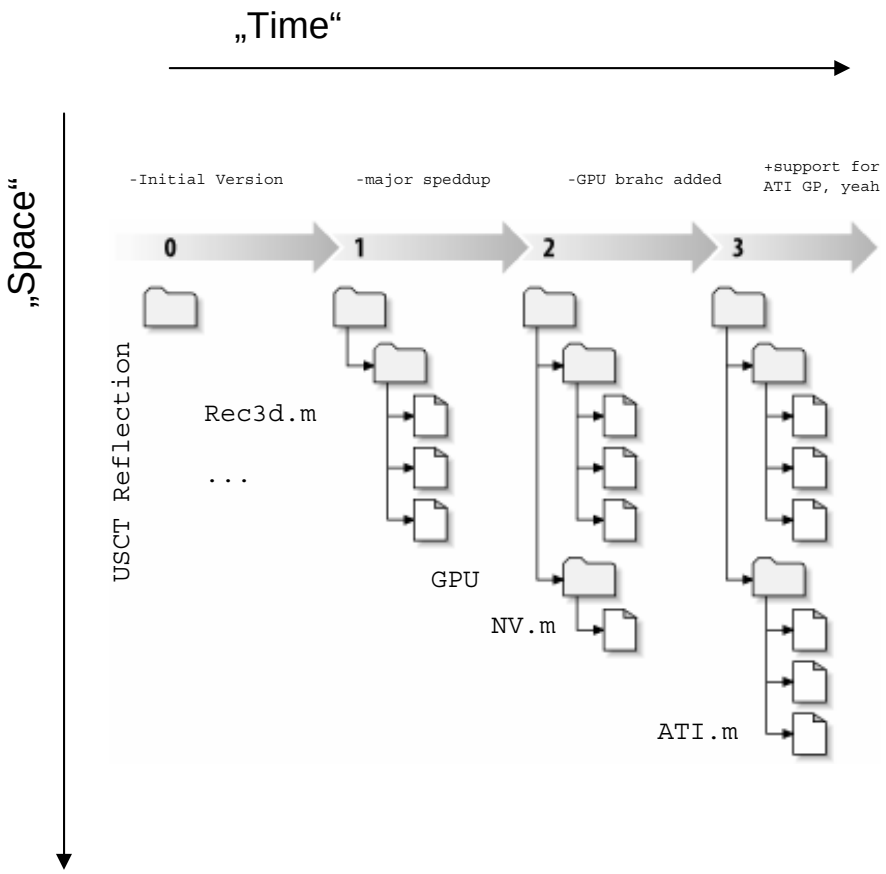
- Data reduction instead of data rate increase
  - Speedup by factor 3
- OR
  - 3 times more measurements possible(!)

- Small potential bandwidth loss, partly compensated by better filter order
- Changed time Offset (only approx. ½ filterlength)

Concept 1: Separation of Concerns

- Solution: separate the concerns, introduce another dimension!
  - Only „spatial aspect“, file and directories, are handled by the OS directory tree
  - „Changes over time“ by some other mechanism -> revision control system
- 2D approach!

- Typical advantages
  - Automated, standardized
  - Meta-data possible (Tags, comments, authors ...)
  - Fine-grained
  - Duplication removed (save space)



Concept 2: Teams

- Typically, software is produced by 1 genius hacker
  - By definition „In sync“
  - Structure clear and perfect
  - No bugs
- Sadly, there are not enough „genius“ hacker available (or projects getting nowadays bigger? ;))
  - A group of software guys has to cooperate in a software project
- Separation of code parts not always perfectly possible, also interface has to exist (and tested)
- Concurrent Code changes happens more likely the more people are involved or the software project size grows
- Some mechanism for handling that situation are required!

Concept 2: Teams - Traditional

- First, traditional approach
  - Exclusive access: lock and free of files
  - Disadavantages
    - Limits developer, discipline required
    - Same as all resource allocators… forgotten frees
    - Workaround happend to often/to easy, forking (copying of locked file) without merging
- Not practical, not working!

Concept 2: Teams - CVS

- Concurrent Versioning System (CVS), first one, defines concepts
- Concept: *„most of the times the overlap is small -> hope for the best!“*

- Consequent -> everyone gets an complete copy („check out“) of everything from some central place („repository“)
  - Allow everyone to do everything on his copy („local copy“), BUT LOG THIS CHANGES

- Integration of changes („check in“)
  - check if changed parts overlapping with changes from some other developer since checkout („update“)
  - If „no“ merge the code („check in“), and hope there is no functional mismatch!
  - If „yes“, cry for manual help („conflict“), but provide tools for resolving

Best practices/consequences

- Usage style
  - Fine grained Check in‘s, do it often, trust the system!
    - Reduces chances for conflicts… really!
    - Makes understanding of changes simpler for other authors
    - Makes Fixing simpler (in the seldome case something broke)
  - But, don‘t expect mircales
    - An version control system is not an Code-review system, nor an statical (or even dynamicla) code analyse tool… it has no clou about the code!
  - Practical, it works
    - even for Million line code projects with hundreds of programmers
    - severe problems are seldom