UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering,
Mathematics & Computer Science

Sparse Array Antenna Signal Reconstruction using Compressive Sensing for Direction of Arrival Estimation

Michiel van Tent Beking

M.Sc. Thesis
April 2016

Supervisors:
dr.ir. G.H.C. van Werkhoven
dr.ir. L. Cifola
dr.ir. M.J. van Bentum
prof. dr.ir. ing. F.B.J. Leferink
prof. dr.ir. F.E. van Vliet
prof. dr.ir. R.N.J. Veldhuis

Telecommunication Engineering Group
Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217

Summary

Direction of Arrival (DOA) estimation in radar applications depends on both the pattern quality and signal to noise ratio of a radar echo. To avoid inaccuracies in the DOA estimation precise knowledge is required for the main beam, and pattern sidelobes should be sufficiently small to suppress signals from other directions. When array antennas are used precise knowledge and small sidelobes are generally obtained by placing the elements at a distance $\frac{\lambda}{2}$ from each other, with λ being the wavelength of the signal. When elements fail or are removed in a traditional dense equispaced $\frac{\lambda}{2}$ array, this is called a sparse array. In a sparse array the antenna beam pattern is degraded due to a broadened main beam with increased side lobes and the presence of grating lobes. Hence more energy is radiated into unwanted directions. It can be said that a dense array with defective elements is some sort of sparse array because the same disadvantages (high sidelobes, degraded antenna patterns) also occur when sparse arrays with element spacings $> \frac{\lambda}{2}$ are used. It is however favorable to use sparse arrays since this can increase the resolution as the same amount of elements can be distributed over a larger antenna aperture. This can reduce in fact the hardware costs since less elements can achieve the resolution of a conventional dense array with more elements.

Traditionally, using array antennas, the DOA is estimated by using beamforming techniques or numerical methods such as MUSIC and ESPRIT. However using these estimation techniques in a sparse array, the degradation of the pattern makes DOA estimation less accurate or even impossible .This requires methods to reconstruct the antenna pattern by other means. The goal of this thesis is to restore the DOA estimation performance by reconstruction of the signal of the failed array elements. For reconstruction techniques from the emerging field of compressive sensing (CS) are used.

A literature study accompanied with Matlab simulations shows that compressive sensing techniques are able to reconstruct the antenna pattern of $\frac{\lambda}{2}$ spaced line array with failed elements. The signal of the *M*-element sparse line array which is reconstructed contains only 25% of the elements of its equivalent fully dense *N*-element $\frac{\lambda}{2}$ spaced array. In general discretization errors dominated the noise errors when the SNR≥20dB. It is possible to reconstruct the original signal when no more than $\frac{M}{\log(N)}$, off grid targets are present at the same time, and when the SNR≥ 10dB and 10 time snapshots are available for reconstruction. When only one time snapshot is available this reduces to approximately $\frac{M}{\log(N)} - 1$ off grid targets with SNR ≥30dB. The accuracy and resolution obtained is in the order of the resolution and accuracy of an equivalent fully dense array, when a grid spacing of 0.25 degrees is used. This makes it promising to use sparse arrays instead of dense arrays, without the disadvantages of a degraded antenna pattern.

The regularized MFOCUSS algorithm, which is an Iterative Regularized Least Squares method, is used for reconstruction. Noise and discretization errors can result in errors when such algorithms are used. The regularization part copes with such errors. In this thesis MFOCUSS is customized to find the optimal regularization parameter using the Generalized Cross Validation technique. This significantly increased the computational load. Simulations with a fixed value for the regularization parameter showed that choosing the regularization parameter between 0.25 and 0.5 on a grid spacing of 0.25 degrees, gives the best results in terms of accuracy and resolution with an SNR of 20dB.

Contents

List of abbreviations

CS	Compressive Sensing
DOA	Direction of arrival
EM	Electro-magnetic
GCV	Generalized Cross Validation (technique)
HPBW	Half power beamwidth
IRLS	Iterative reweighted least squares
MFOCUSS	Multiple Focal Underdetermined System Solver
MMV	Multiple measurement vectors
MSE	Mean square error
RCS	Radar cross section
RX	Receive
SNR	Signal to noise ratio
ULA	Uniform line array

List of symbols

Symbol	Description
$\mathbf{a}(\theta)$	[Nx1] array steering vector corresponding to the direction θ.
AF	Array Factor
C	constant (close to one)
D	antenna dimension
K	number of incoming targets/non-sparse components
M	number of working elements in the sparse array
N	number of elements of the fully dense array
$\mathrm{P}_{\mathrm{y}}(\theta)$	[N_s x 1] vector containing the angle spectrum of the solution
$\boldsymbol{s}$	[K x 1] vector representing the sources
T_s	number of time samples
$\mathbf{W}$	[N x T_s] Noise matrix
$\mathbf{W}_k$	Weighing matrix used in MFOCUSS
$\mathbf{X}$	[N x T_s] original signal matrix
$\mathbf{Y}$	[M x T_s] (Compressed) measurement matrix
$\mathbf{Z}$	[N_s x T_s] solution matrix
$\mathbf{w}$	[N x 1] Noise vector
$\mathbf{x}$	[N x 1] original signal vector
$\mathbf{y}$	[M x 1] (Compressed) measurement vector
$\mathbf{z}$	[N_s x 1] sparse solution vector
$\hat{\boldsymbol{z}}$	[N_s x 1] reconstructed source vector
λ	Regularization parameter used for the MFOCUSS algorithm
λ_w	Wavelength of the transmitted and received reflected signal
$\Delta\phi$	Phase difference between elements
θ_B	Half power beamwidht
θ	Scan angle of the array i.r.t. the boresight direction
$\mathbf{\Phi}$	[M x N] measurement matrix or in general the sensing matrix
$\mathbf{\Psi}$	[N x N_s] scan angle matrix or in general the sparsity basis matrix
$\mathbf{\Theta}$	[MxN] observation matrix
$\lVert\cdot\rVert_{\boldsymbol{p}}$	The ℓ_p norm**,** with p the number of the norm (e.g. 0, 1, 2 of F for the Frobenius norm)
$.^{\dagger}$	Take the pseudo-inverse
$.^{\boldsymbol{H}}$	Take the Hermitian/complex conjugate transpose

1 Introduction

This thesis is the result of the work performed for a master assignment in Electrical Engineering at the Telecommunication Engineering Group of the University of Twente. The work was done externally at Thales Nederland in Hengelo, the Netherlands. In this thesis the relevant information and obtained results are presented. This thesis starts with background information, which serves as an introduction to the subject of the thesis.

1.1 Background

Direction of Arrival (DOA) estimation is an important application of Radar systems. Besides the traditional beamforming method [1], there are several other ways to determine the DOA. Popular examples of other ways are more modern high resolution sub-space methods such as MUSIC and ESPRIT [1], [2]. Although they can have advantages of an increased resolution they also suffer from the fact that the number of incoming targets must be known a-priori and they need a lot of time snapshots to work properly [1]. This makes them less suitable for the use in a radar system, where in general the number of incoming targets is not known and the number of time snapshots is limited.

DOA estimation with the traditional beamforming method (and also with numerical methods such as MUSIC and ESPRIT) relies heavily on a good a priori known antenna pattern. If for some reason an element in a phased array antenna does not perform according to its specifications or even fails, the antenna pattern degrades. The level of degradation depends on the number of failing elements. When in a dense array elements are removed randomly on purpose, one speaks of a sparse array since the elements become thinly dispersed as more and more elements are removed. Poor performing, failing or missing elements in a dense array cause a decrease in the gain, a wider main beam and an increase in the peak sidelobe level. This results in a reduction of the total maximal output power and directivity, and increased power in the sidelobes. Overall this results in a heavily degraded antenna pattern and a reduction of the ability to estimate the DOA correctly. The degraded antenna pattern is the main reason sparse arrays are in general not used in radar systems.

When the degraded antenna pattern of a sparse array can be reconstructed the DOA can be estimated correctly again. Besides the degradation of the antenna pattern, using sparse arrays comes with advantages of:

1. An enhanced angular resolution while using the same amount of elements that are normally used in a smaller dense array. The angular resolution of the antenna is mainly determined by its size, hence this results in systems that have a better angular resolution as a dense array since the elements needed can be spread out over a larger dimension
2. A reduction in mutual coupling effects due to the larger element spacing in sparse arrays.

When it is possible to reconstruct the original signal and antenna pattern the advantages of an enhanced angular resolution and less mutual coupling remain, while the disadvantages are resolved. Besides that one can also create an advantage in robustness against failures. When many elements in a dense array are faulty the array behaves as if it were a sparse array. In an operational phased array antenna it is not always desirable or possible to repair or replace faulty elements. In land based or naval based systems downtime is often not desired and in airborne systems maintenance is simply not possible while in flight. In the extreme case of phased arrays on satellites maintenance or replacement is (nearly) impossible after the satellite is space-borne. This asks for alternative methods to reconstruct the performance or antenna pattern of phased arrays in the case of faulty elements without replacing the faulty elements. This increases the effective usefulness and dependability of phased array antennas.

Being able to reconstruct the signal and to use sparse arrays without loss of performance makes it also possible to anticipate on the demands of users of phased array antennas. Due to worldwide defense budget cuts there is more and more demand for low cost, but high performance technology. Another trend due to budget cuts is the trend towards having less and less people on board of (navy)ships. Also training courses are shortened and crew is more and more “learning on the job”. This has also consequences for Radar and other antenna systems; more and more has to be automated and less and less maintenance must be necessary. Being able to use phased arrays without significant loss of performance in case of failures responds to the wish for more automated systems and decrease the number of necessary (maintenance) crew members.

An emerging technique to reconstruct missing data is “Compressive Sensing” (CS) or “Compressive Sampling”. Although the field of CS is emerging the last decade, there are not many applications found in literature in which it is used to reconstruct signals of sparse array antennas.

1.2 Research objectives

The main research goal of this thesis is to determine whether it is possible to reconstruct the received signal of a sparse or faulty array with CS and, using this reconstruction, to estimate the DOA of incoming targets with this reconstruction. The aim is to estimate the DOA with the same accuracy and resolution of an equivalent dense array of the same size. Since CS comes with several demands and constraints, it is also questioned how these demands and constraints translate into constraints and demands in terms of:

1. The accuracy and resolution;
2. Required SNR;
3. The number of simultaneously present targets that can be correctly estimated.

To answer these questions a literature study was done to familiarize with the techniques of CS and DOA estimation. Then the obtained knowledge was used to simulate a sparse array in Matlab, and CS was applied to test the performance of the reconstruction.

1.3 Outline of the thesis

This thesis is organized as follows. After this introduction, in chapter 2 the relevant theory of phased array antennas is given in such a way that it is prepared for the CS signal reconstruction method. In chapter 3 the CS reconstruction method is explained and readily applied on the DOA estimation problem. After that the explanation and implementation of the reconstruction algorithms in Matlab is discussed. The simulation set up in Matlab concludes chapter 3. In chapter 4 the results are presented with a short analysis and discussion per section. Conclusions and recommendations can be found in chapter 5.

2 Phased arrays Antennas and Direction of Arrival Estimation

Knowledge of phased array antennas and radar systems is necessary if reconstruction techniques are to be applied to these systems. A phased array antenna is a directive antenna made up of a number of individual antennas or radiating elements. The phased array antenna has the advantage of being able to steer its beam electronically by changing the phase of each element. This means that the beam can be steered rapidly from one angle to another one without the need for mechanically positioning the antenna. The antenna can be used for both transmitting and receiving energy. The estimation of the DOA uses the received energy, thus in this thesis the focus lies on the receive (RX) antenna and RX part of the radar system. This chapter will shortly address and explain the most important theory and techniques used in a modern phased array radar system which are needed for reconstruction. The treated theory will be definitely non-exhaustive and treated in such a way that it is almost directly applicable to the framework needed for reconstruction.

2.1 RADAR preliminaries

The principle of operation of a radar system is based on transmission of an electromagnetic wave and its reflection from an object. When the object is an object of interest for the radar operators, such as hostile airplanes, missile or ships the object is called a target. The reflection or “echo” can be used to determine the range and (radial) velocity (using pulse train and Doppler effects), and the direction of arrival of the targets. The range is determined by the time difference between sending the pulse and receiving the echo. The (radial) velocity is determined by the Doppler effect that changes the frequency of the transmitted signal when it is reflected by a moving target, back to the radar. The direction can be measured from the direction towards which the antenna is pointing when the echo is received. The estimation of the direction of the target is depending on the directivity and beamwidth of the RX antenna. Directivity, described as the ability of the antenna to concentrate the transmitted energy in a desired direction, is a parameter determined by the size of the antenna. The beamwidth is the maximal diameter of the main beam of the antenna. The directivity and the beamwidth are both a function of the size of the antenna, making the accuracy of the direction measurement dependent on the size of the RX antenna.

The amount of energy that is reflected by the target is determined by its Radar Cross Section (RCS). The RCS depends on the geometry and material of the reflecting object and on the used frequency. Hence it is possible that different objects at the same distance from the radar reflect a different amount of the incident energy. This results in different received power levels back to the radar RX antenna. When the reflected energy is very low the antenna might not be able to detect it because it is below the noise floor of the receiver. The Signal to Noise ratio (SNR), usually expressed in dB, gives the ratio between the received signal level and the noise level. Hence a certain SNR is required for detection of a target. Integration of the pulses can increase the SNR. Normally in radar there is integration gain through pulse compression and Doppler integration. Additionally there is beamforming gain which combines the energy received by multiple elements. Normally after pulse compression, Doppler integration and beamforming the SNR must be higher than 0dB to be able to detect small targets. In this thesis, using CS, it is assumed that pulse compression and Doppler integration are already applied, but beamforming is not yet applied. Thus the SNR must be higher than 0dB after only pulse compression and Doppler integration, meaning that it is expected that very small targets cannot be detected.

In this thesis the objects that reflect energy back towards the radar are all called “targets” for simplicity. When is spoken of a “signal”, it is referred to the signal received by the (individual elements of an) array antenna.

2.2 Antenna pattern of phased array antennas

The radiation pattern of a phased array antenna is determined by the phase and amplitude of the signal transmitted or received by the antenna elements and the relative positions of the elements [3]. This can be explained as follows: consider N elements placed in a straight line in one dimension, equally spaced a distance *d* apart as in Figure 1. The individual elements are equal point sources and assumed to be isotropic radiators, i.e. they have a uniform response in every direction. This configuration is usually referred to as a Uniform Line Array (ULA). When all the individual elements transmit the same signal the result in the far field is a plane wave transmitted in the orthogonal direction with respect to the array surface (broadside direction). By controlling the progressive phase difference between the elements this plane wave can be scanned to a certain direction. From the simple geometry in Figure 1 it follows that the difference in path length between adjacent antenna elements for scanning the beam towards an angle θ with respect to the normal to the antenna, is $d\sin\theta$. This difference in path length between the adjacent elements results in a phase difference of $\Delta\phi = 2\pi\frac{d}{\lambda_w}\sin\theta$, with λ_w the wavelength of the transmitted signal.

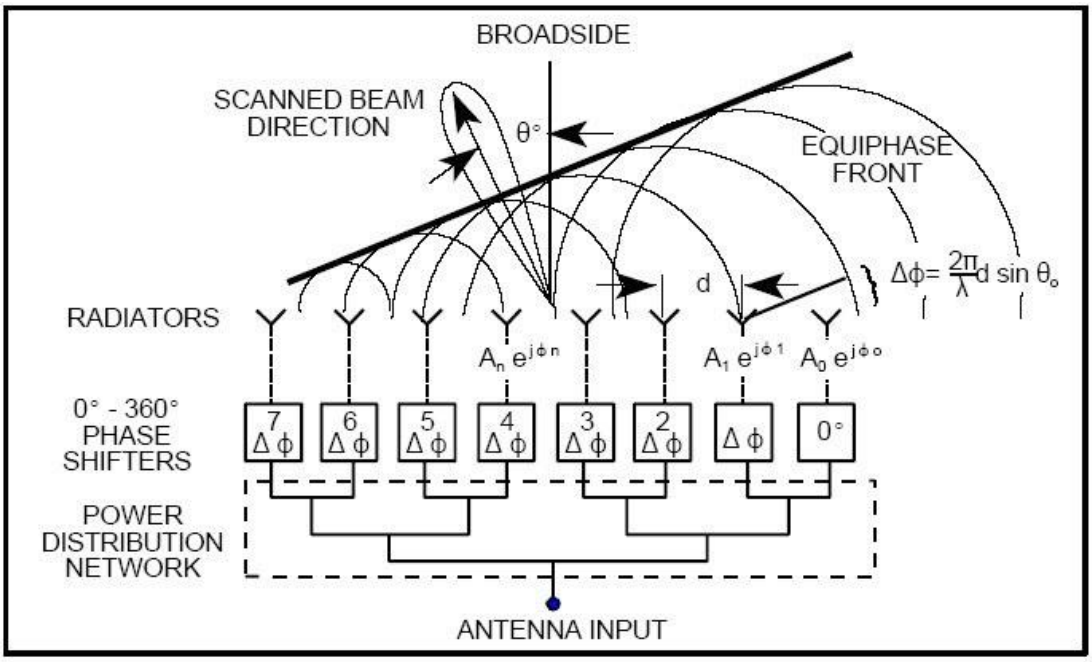

Figure 1: 7 element receiving linear array.

When the amplitude *A* of the transmitted signal is taken at unity, every element transmits a signal which is equal to $e^{j\alpha(N-1)}$, with $\alpha = \Delta\phi = 2\pi\frac{d}{\lambda_w}\sin\theta$ i.e the phase difference between the elements. This means that the direction of the transmitted signal depends on progressive phase shift between the elements, and the distance d between elements.

2.2.1 Array factor, side lobes and grating lobes

It is well known in phased array theory that the far-zone electro-magnetic (EM) field pattern of an array of identical elements with uniform amplitude and spacing, is equal to the product of the EM field of a single element, at a selected reference point (usually the origin) and the array factor (AF) of that array [2]:

$$\mathbf{E}(\text{total}) = [\mathbf{E}(\text{single element at reference point})] \times [\text{array factor}]. \tag{1}$$

Note that such an array of identical elements, all with equal magnitude and spacing and thus progressive phase shift, is referred to as a ULA. Besides the assumption of identical element patterns, there are also no other errors assumed. The far zone field in this case is defined as $r = \frac{2D^2}{\lambda_w}$ with *D* the antenna dimension. Since the radiating elements are assumed to be point sources, the AF in this case is simply the summation of the signals $e^{j\alpha(n-1)}$ at each element:

$$\text{AF} = 1 + e^{j\alpha} + e^{j2\alpha} + \cdots + e^{j(n-1)\alpha}. \tag{2}$$

To decrease sidelobes, it is common to apply an amplitude taper on the array by applying weighing coefficients on the amplitudes. Usually also the tapering or weighing coefficients $\omega_n$are included in the array factor. This makes the AF:

$$\text{AF} = \sum_{n=1}^{N} \omega_n e^{j(n-1)\alpha}. \tag{3}$$

For this thesis there is no amplitude tapering assumed: ω_n=1. This makes it that the AF can also be written as [2]:

$$AF = e^{j\left(\frac{N-1}{2}\right)\alpha} \left[\frac{\sin\frac{N}{2}\alpha}{\sin\frac{1}{2}\alpha}\right] \tag{4}$$

The first factor is basically a phase shift $\left(\frac{N-1}{2}\right)\alpha$. When the reference element in the array was chosen in the middle instead of on the left hand side, this phase shift would be zero. This reduces the AF for an ULA to [2]:

$$(AF) = \left[\frac{\sin\frac{N}{2}\alpha}{\sin\frac{1}{2}\alpha}\right] \tag{5}$$

The maximum value of (5) for the AF is equal to N. The AF is solely a function of the geometry of the array and the excitation phase. By varying the separation distance *d* between the elements and/or the excitation phase α between the elements, the characteristics of the array factor and of the total field of the array can be controlled [2]. The field intensity pattern has zeros when the numerator of equation (5) is 0. This occurs when $\sin\frac{N}{2}\alpha = \sin\left[N\pi\frac{d}{\lambda_w}\sin\theta\right] = 0, \pm\pi, \pm2\pi, \dots, \pm n\pi$ where n is an integer. The denominator on the other hand is zero whenever $\frac{d}{\lambda_w}\sin\theta = 0, \pm\pi, \pm2\pi, \dots, \pm n\pi.$

The maximum in the field pattern occurs when $\sin\theta = \pm\frac{n\lambda_w}{d}$ [3]. In this the maximum at $\theta = 0$ degrees defines the main beam of the antenna pattern. The other maxima are called grating lobes and have the same magnitude as the main beam. They are undesirable because they can cause ambiguities by being mistaken for responses belonging to the main beam.

2.2.2 Scanning the main beam

When the element spacing is fixed at known distances *d,* this means that the progressive phase shift of the signals at each element determines the scan angle of the main beam. Thus by controlling the progressive phase difference between the elements, the main beam can be squinted in the desired direction. For the total array antenna the progressive phase shift for each element can be placed into a so called "steering vector". Hence the steering vector represents the relative phase shifts for the incident far-field waveform across the array elements and is described as $\mathbf{a}(\theta) = \left[1\; e^{j\alpha} \ldots e^{j\alpha(N-1)}\right]$. Now the main beam can be scanned to a certain angle simply by applying the steering vector to the signal and thus add progressive phase shift to the signals for each element. This concept is called beamforming.

Since maxima occur when $\sin\theta = \pm\frac{n\lambda_w}{d} = 1$, scanning the main beam unfortunately has also an effect on the grating lobes. For a beamforming array grating lobes can be avoided by choosing a spacing between the elements that satisfy:

$$\frac{d}{\lambda_w} \leq \frac{1}{1 + sin\theta_{max}} \tag{6}$$

In practice this means that the element spacing *d* is often fixed at $\frac{\lambda_w}{2}$ to avoid grating lobes in the "operating" area between -90 and 90 degrees completely. (When the element spacing is $\frac{\lambda_w}{2}$ a grating lobe will appear at -90 degrees when the main beam is scanned to 90 degrees. Although practical phased arrays are usually limited to a scan angle of 60 degrees, this still means an element spacing of 0.54 λ_w is needed according to equation (6). Hence the spacing is usually limited to $\frac{\lambda}{2}$.) The occurrence of grating lobes for spacings $> \frac{\lambda_w}{2}$ in the visible area is one of the main reasons why arrays with a spacing larger than $\frac{\lambda_w}{2}$ are not used in phased array radar applications.

There are also secondary maxima, called sidelobes, which occur approximately when the numerator of equation (6) attains its maximum value. This results in a magnitude of approximately 13.2dB down from the maximum of the main lobe, assuming uniform illumination for the array [3]. A sidelobe is thus a local maximum in the antenna pattern outside the main beam. There are two important reasons to keep the sidelobes as low as possible. In the first place a target with a large reflection/RCS in a sidelobe can result in a false detection in the main lobe because the target strength is above the threshold due to the high sidelobe level. Secondly a weak target in the main beam can be masked by a strong target in a sidelobe.

In Figure 2 the sidelobes and grating lobes are shown for arrays with element spacings λ_w and $\frac{\lambda_w}{2}$ for a scan angle of 0 and 30 degrees respectively. It can be seen that for the $\frac{\lambda_w}{2}$ spaced array no grating lobes occur. For the λ_w spaced array however grating lobes are visible at the edges when the array is scanned to 0 degrees and in the visible area at -30 degrees when the array is scanned towards 30 degrees. This makes target detections not unambiguous anymore.

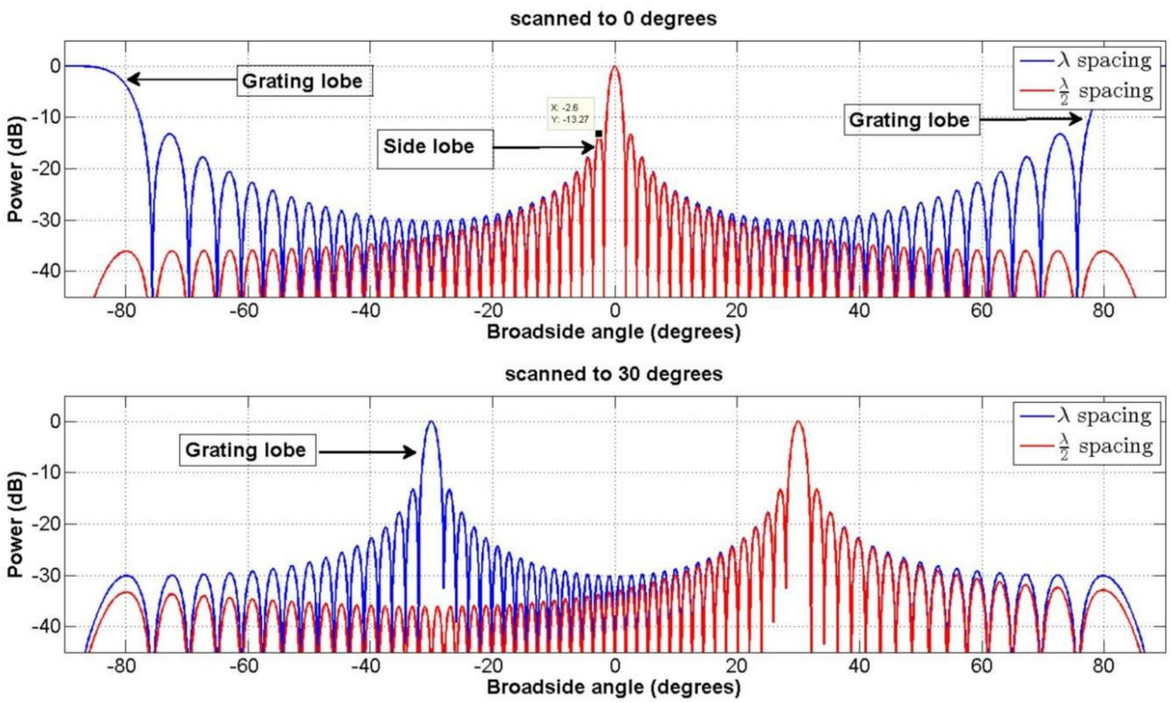

Figure 2: Grating lobes and side lobes for element spacings of $\frac{\lambda}{2}$ and λ for ULAs with an equal length.

2.2.3 Effects of scanning the main beam

As the beam of a phased array is scanned towards an angle θ from broadside the beamwidth of the main beam increases with $\frac{1}{\cos\theta}$. This is valid until the angle θ gets large. In practical systems this means that at a scan angle 60 degrees off broadside the beamwidth of the array increases with about a factor 2. Another effect that occurs while scanning the main beam off broadside is that the antenna gain also decreases with a factor $\frac{1}{\cos\theta}$. When the scan angle gets larger than 60 degrees other effects such as mutual coupling effects can increase significantly, making the increase in beamwidth and decrease in antenna gain from this simple theory not valid anymore. Also the sidelobes increase more than expected from simple theory [3]. This is the main reason why in practical systems, and in this thesis, the maximum scan angle is limited to 60 degrees.

2.2.4 Element failures & effects on the antenna pattern

Failure of a sufficient number of antenna elements can seriously degrade the performance of a (low side lobe) array antenna [3]. The beamwidth of an array antenna is determined by the aperture size. The gain and sidelobe levels are determined by the number of elements that remain. Hence a thinned or sparse array will have about the same beamwidth as a filled half-wavelength array, but its gain will be reduced in proportion to the number of elements removed [3].

Definitions of sparse or thinned arrays differ in literature but in general an array is called sparse or thinned when the element spacing of half-wavelength (to avoid grating lobes when steering between -90 and 90 degrees) is not respected anymore. Thus when the elements are spaced unequally and on average spaced much more than half-wavelength the array is said to be thinned or sparse [3]. Thinning or making an array sparse will however produce serious undesirable changes is the antenna pattern when the thinning is too high or the array becomes too sparse. The gain will be reduced and there will be high peak and average sidelobes. A conventional half-wavelength spaced array will have almost all of its radiated energy within the main beam, only a few percent of the radiated energy will appear in the sidelobes. When the array is thinned more and more and becomes sparser the reverse is true. Too much energy ends up in the sidelobes and is practically wasted. An array which is thinned 90 percent (e.g. 90 percent of the elements fail) might have about 90 percent of its energy in the sidelobes [3]. The effect on the antenna pattern of letting ≈90 percent of the elements fail is shown in Figure 3, which show that such a large percentage of missing elements results in an antenna which has barely any directivity.

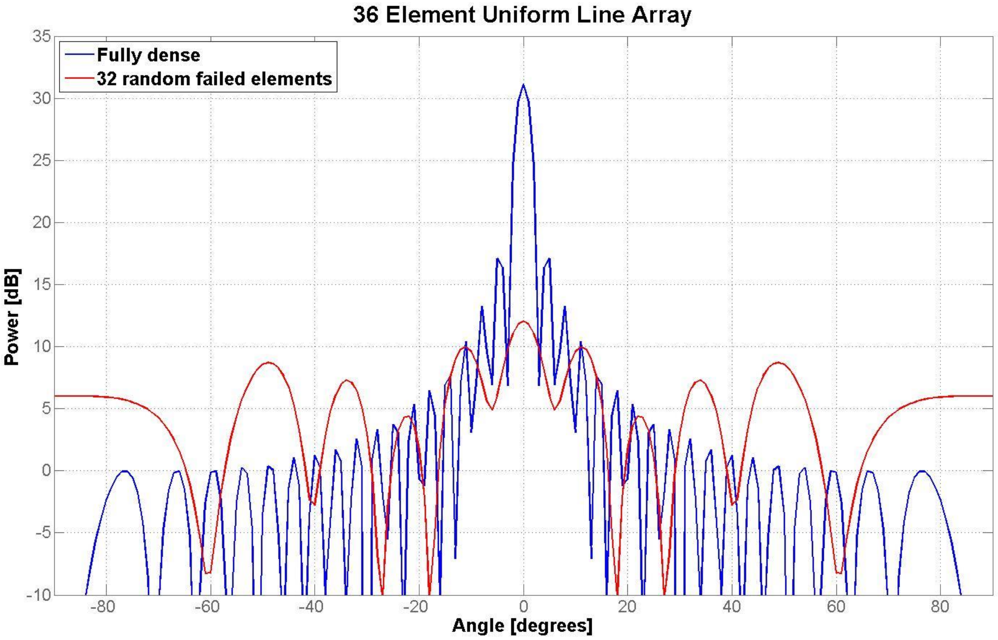

Figure 3: Effect on the antenna pattern of thinning a 36 element ULA with approximately 90 percent.

The $\frac{\lambda_w}{2}$ uniform spacing in an ULA makes it that there is an antenna pattern with one main beam and relatively low (-13dB) sidelobes. When elements in the ULA fail the uniform spacing is interrupted, which gives rise to sidelobes and grating lobes. This can be seen in Figure 4 where the antenna pattern of a 36 element ULA and a 36 element ULA with 2 random failed elements is shown. The maximum peak sidelobe level is still the same (-13 dB), but the removal of just 2 elements causes an increase of the sidelobe at other directions, causing an increase of the average sidelobe level. This makes it possible that large targets at this angle may appear in the main beam.

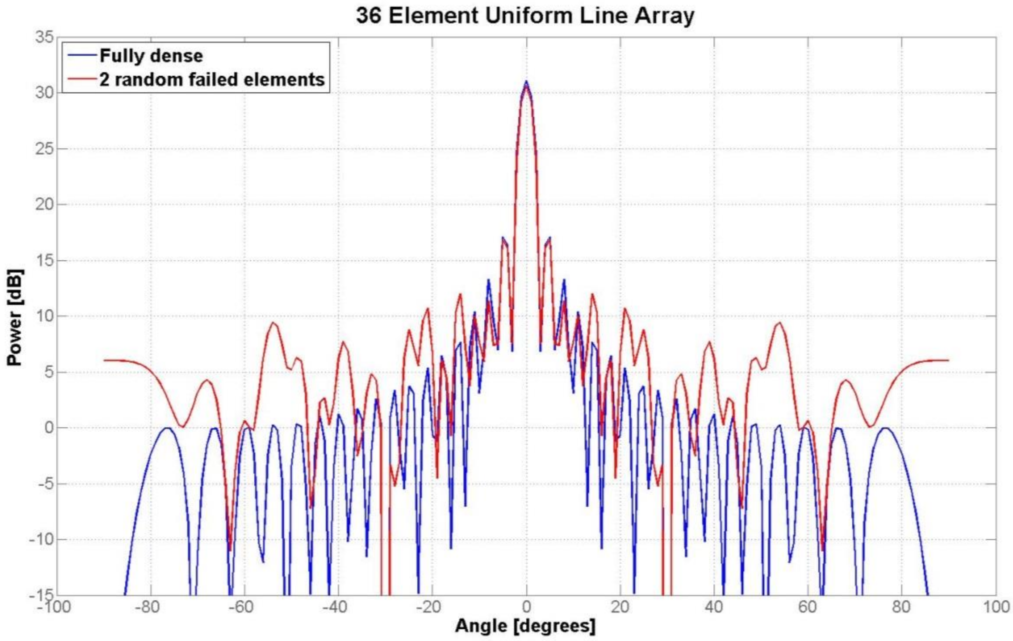

Figure 4: the antenna pattern of a 36 element ULA and a 36 element ULA with 2 random failed elements.

Another negative side effect of failing elements is the decrease in gain due to the fact that there are simply less elements available which can radiate energy into the desired direction. Gain is a measure of the ability of an antenna to concentrate the transmitted energy in a particular desired direction [3]. In Figure 5 the effect on the gain can be seen at the main beam (at 0 degrees). The ULA with 6 failing elements has a loss in gain of approximately 3dB. This means that half the power is radiated in the desired direction compared to the fully functional ULA.

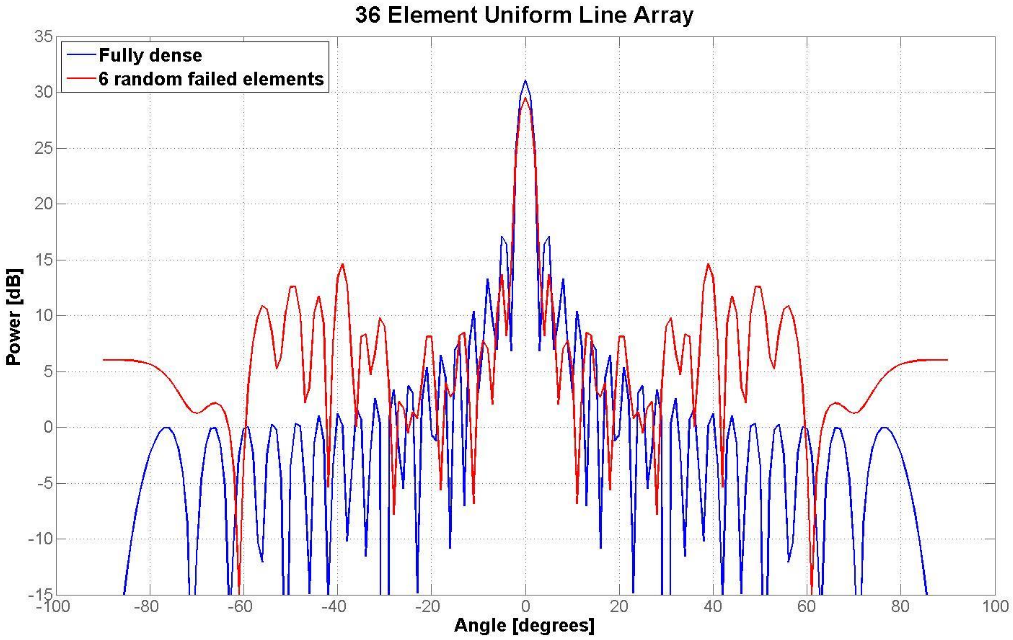

Figure 5: 36 element ULA vs a 36 element ULA with 6 failing elements. The loss in gain is approximately 3dB.

When more elements fail in the array the increase in sidelobes and decrease in gain gets worse. An example is shown in Figure 6 where it is shown what is the effect when two third of the elements in a

36 element ULA fail. It can be seen that the antenna pattern looks more like a sequence of sidelobes than a main beam with some sidelobes. Such patterns are quite unsuitable for DOA estimation using beam patterns (monopulse etc.). Also numerical methods (such as MUSIC, ESPRIT etc.) will have problems with such patterns. CS however, might be able to do a better job.

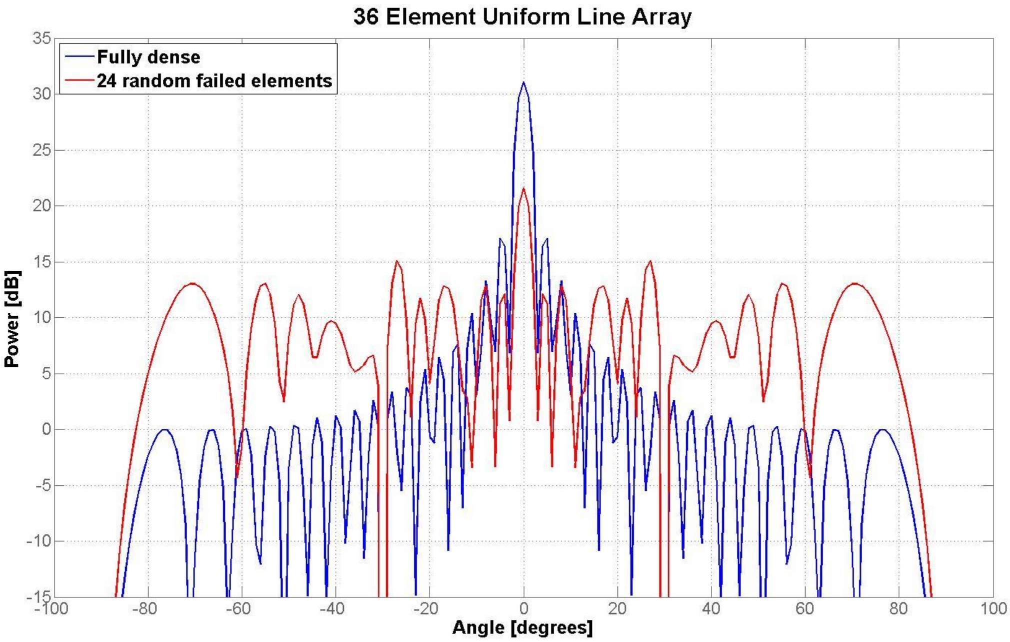

Figure 6: 36 element ULA versus a 12 element non uniform array with the same aperture.

In Figure 6 the array with 24 failing elements still has the same length as the fully dense ULA, since the 2 outermost elements are still working. When the outermost elements are also failing this means that the length or (1 dimensional) antenna dimension *D* is decreased.

The antenna dimension *D* has a direct impact on the resolution of the antenna according to the relation:

$$\theta_{3dB} \cong \frac{\lambda_w}{D} = \frac{\lambda_w}{N\frac{\lambda_w}{2}} = \frac{2}{N} \text{ [rad]} \tag{7}$$

Where θ_{3dB} is the half power beamwidth (HPBW) of the antenna. The HPBW is the angular range of the antenna pattern in which at least half of the maximum power emitted.

The effect of reducing the antenna dimension *D* can clearly be seen in Figure 7 where the effect of a decrease in aperture is shown for an 18 element ULA and a 36 element ULA. According to equation (7), the 18 and 36 element arrays have an HPBW of approximately 5.7 and 2.8 degrees respectively. This can clearly be seen in Figure 7: the main beam of the 18 element array have a much broader main beam due to the fact that the aperture is halved compared to the 36 element ULA.

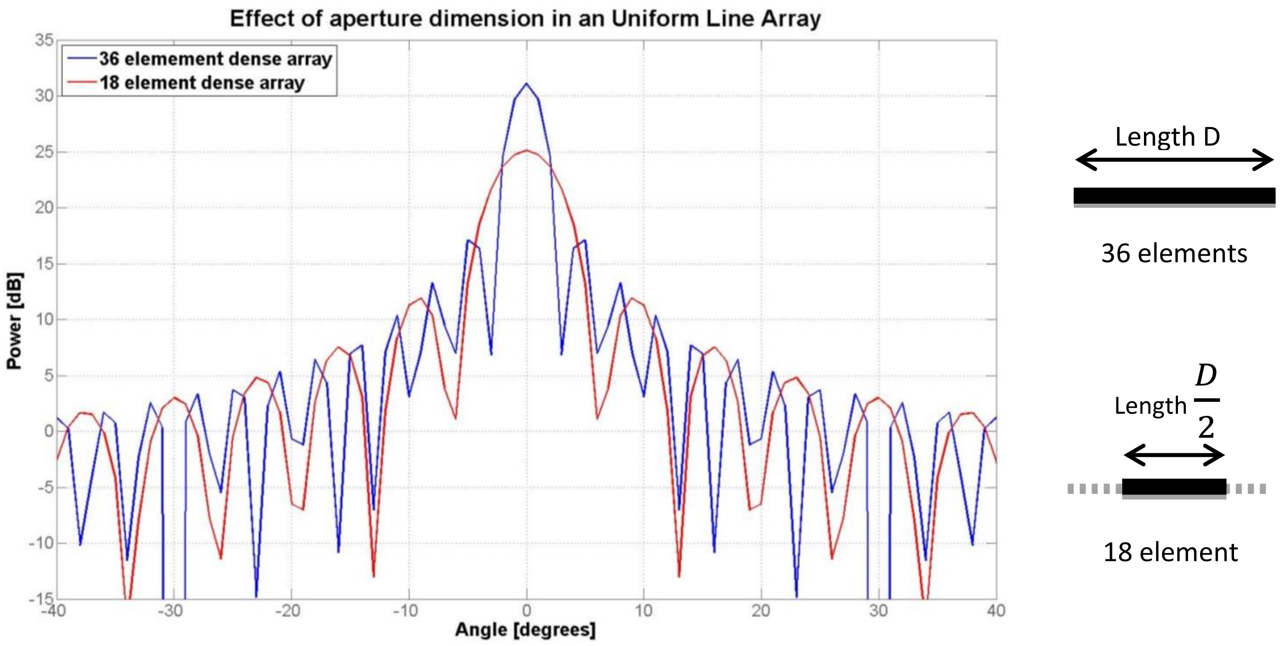

Figure 7: Effect of a decrease in aperture on the half power beamwidth.

The purpose of reconstruction in this thesis is to be able to obtain the HPBW and same sidelobe and grating lobe properties of a fully dense array using only the signals received by the elements of the equivalent sized sparse array. This should then result in an improved resolution considering the number of elements used.

2.3 DOA estimation using a phased array antenna

The concept of beamforming can also be used in reverse to estimate the DOA of targets reflections impinging on the array. Hence at time instant *t*=1,2...,T with *T* the total number of snapshots, the received signal of an *N* element array can be written as a [*N*x1] column vector **x**(*t*) of the form:

$$\mathbf{x}(t) = \sum_{k=1}^{K} \mathbf{a}(\theta_k) s_k(t) + \boldsymbol{w}(t) \tag{8}$$

where:

- $s_k(t)$, *k*=1,2,...,*K*, is a narrow planar wave front impinging on the array from a direction θ_k;
- *K* is the number of source wave fronts;
- **w**(*t*) is an [*N* x 1] vector representing the additive noise at the array;
- **a**(θ_k) is an array steering vector of size [*N* x 1] corresponding to the source from a direction θ_k.

This received signal vector **x**(*t*) of equation (8) can now be expressed as:

$$\mathbf{x}(t) = \ \mathbf{As}(t) + \ \mathbf{w}(t) \tag{9}$$

where **A** = [$\mathbf{a}(\theta_1)\mathbf{a}(\theta_2) \ldots \mathbf{a}(\theta_K)$] is an [*N* x *K*] matrix of steering vectors (with $\mathbf{a}(\theta) = \left[1\ e^{j\alpha} \ldots e^{j\alpha(N-1)}\right]$ and **s**(t) = [$\boldsymbol{s}_1(t)$,$\boldsymbol{s}_2(t) \ldots \boldsymbol{s}_\mathrm{K}(t)$]$^\mathsf{T}$ is a [*K* x 1] vector representing the sources.

Since it is generally not known in radar from which directions targets arrive, the vector **s** is unknown, but there are several ways to find them. With traditional sum beamforming one can "search" for constructive (and destructive) interference between the signals received at each element of the array. This is done simply by discretizing the area of observation into N_s angles and then scan the main beam to every N_s angles and look for constructive interference. This can be mathematically described as: to find the DOA of incoming targets, look for constructive interference in all direction using the steering vector $\boldsymbol{a}(\theta)$ for N_s values of θ. This result in a [N x N_s] scan angle matrix $\boldsymbol{\Psi}$:

$$\boldsymbol{\Psi} = [\mathbf{a}(\theta_1)\,\mathbf{a}(\theta_2)\ldots\mathbf{a}(\theta_{Ns})], \qquad \textbf{(10)}$$

with $\theta_1, \theta_2, \ldots, \theta_{Ns}$ the set of angles to scan (e.g. grid points on the scan grid). Now let the received signal from equation (8) be an N-dimensional vector **x**:

$$\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix} \in \mathbf{R}^N \quad \text{of real numbers where } N \text{ is large.} \qquad \textbf{(11)}$$

When the received signal is multiplied with the scan angle matrix $\boldsymbol{\Psi}$ this results in a scanned beam pattern in which the DOAs of the targets are clearly visible. This scanned beam pattern of the received signal that is used for DOA estimation can be expressed in a similar way as equation (9):

$$\mathbf{x}(t) = \boldsymbol{\Psi}\mathbf{s}(t) + \mathbf{w}(t) \qquad \textbf{(12)}$$

An example of such a scanned beam pattern is shown in Figure 8. In this angle spectrum 3 targets arrive on a 36 element ULA at -30, -10 and 20 degrees. The element spacing d of the ULA is $\frac{\lambda_w}{2}$. The different amplitudes in the angle spectrum indicate that the targets have different strengths. In radar this indicates a different RCS of the target since the difference in range is usually already accounted for by the difference in delay. Note that in Figure 8 the amplitude excitation of the elements in uniform (no tapering applied) which results in a first sidelobe level or 13dB, as explained in the section Array factor, side lobes and grating lobes.

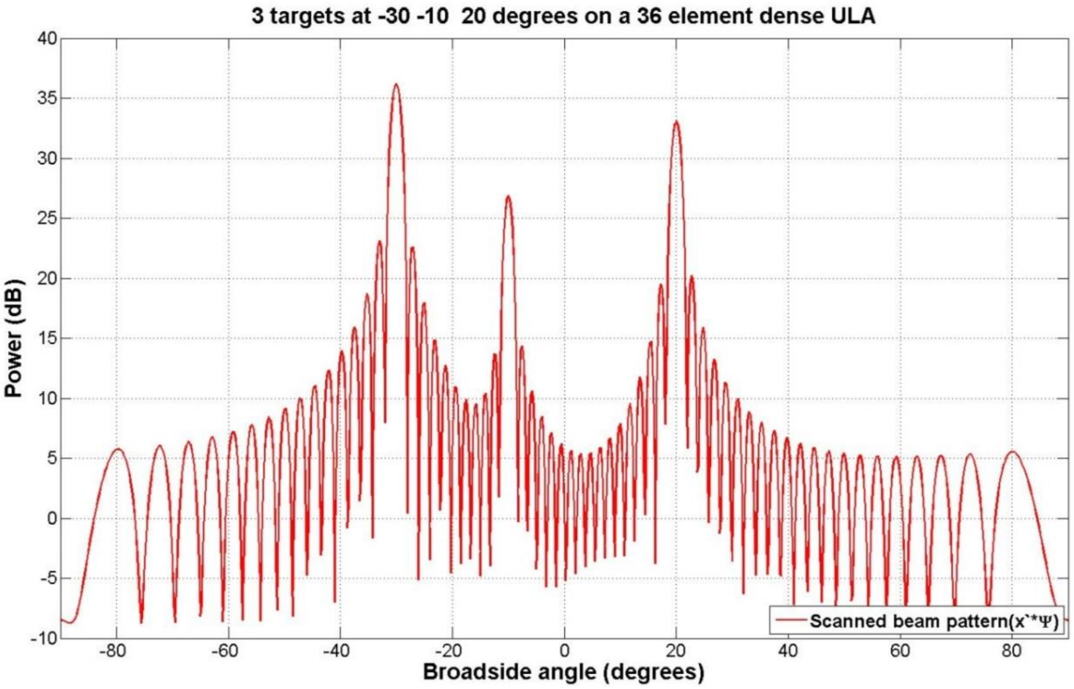

Figure 8: Scanned beam pattern of a 36 element array on which 3 targets with different RCS arrive. Element spacing d=$\frac{\lambda}{2}$.

2.4 Accuracy, Resolution and Cramér-Rao bounds

Two important performance parameters of DOA estimation are the angular resolution and the accuracy. The accuracy has to do with the error in the estimation of the DOA of a single target at a certain SNR. The theoretical accuracy with which an ULA can measure the DOA according to [3] is:

$$\sigma_{\sin\theta} = \frac{\sqrt{3}\lambda_w}{\pi D\sqrt{\frac{2E}{N_0}}} = \sigma_{\sin\theta} = \frac{\sqrt{3}}{\pi}\frac{\frac{\lambda_w}{D}}{\sqrt{\frac{2E}{N_0}}} = \frac{\sqrt{3}}{\pi}\frac{\frac{\lambda_w}{D}}{\sqrt{2}\sqrt{SNR}} \tag{13}$$

When it is possible to restore the degraded antenna pattern it is important to compare the obtained angular resolution and accuracy with the theoretical lower bounds. This tells something about the performance of the system for DOA estimation.

The angular resolution is the minimum angular separation at which two equal targets can be separated. That again means that two identical targets at the same distance are only resolved in angle if they are separated by more than the antenna HPBW.

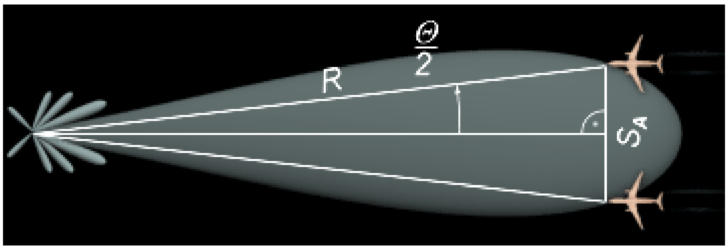

Figure 9: The angular resolution is the minimal distance S_A required between two targets at range R from the antenna at which they can be distinguished from each other. (source www.radartutorial.eu)

The angular resolution of an antenna can be determined from the simple geometry of Figure 9, and depends on the beamwidth of the used antenna and the range of the targets as:

$$S_A \geq 2R\sin\frac{\theta_{3dB}}{2}. \tag{14}$$

This resolution, which is basically the beamwidth, can be increased in the processing of the signals, exploiting high SNRs and phase differences for instance, but it is generally accepted that two equal targets can be resolved in angle when they are separated at least eight-tenths of a beamwidth [3].

Since the SNR has to be above the system noise floor for detection, it is favorable to have an equation for the angular resolution which also shows the dependence on the SNR. This in fact the Cramér-Rao Bound for the resolution of two targets (separate two targets) [4]:

$$\Delta\sin\theta = \frac{1.76\frac{\lambda_w}{D}}{\sqrt[6]{SNR}}. \tag{15}$$

The Cramér-Rao bound is calculated for complex estimators. For the angular position only the real part of the estimator is used which makes the lower bound a factor $\frac{1}{\sqrt{2}}$ better. The Cramér-Rao bound is a theoretical lower bound on the estimation accuracy, given the measurements. It is a theoretical bound, it is not guaranteed there is an algorithm or method that can achieve the resolution obtained through the Cramér-Rao bound. Using reconstruction methods changes the measurements and estimation uncertainties, hence the Cramér-Rao bound of (15) is used to compare the results and to see if the resolution is of the same level as the theoretical minimum of a conventional fully dense array.

The most common way to increase the resolution is to increase the antenna dimension (see equation (7),(14) and (15)). This means increasing costs because more elements are needed ($N = \frac{D}{\lambda_w/2}$) to avoid grating lobes. A non-uniform element spacing can avoid this, but this gives rise to higher sidelobes, which can give false detections. This is schematically shown in Figure 10. When a dense array is thinned the sidelobes get higher and the gain decreases. It should be noted that although the decrease in gain is an important loss of performance, this thesis does not focus on the actual decrease in gain, but more on the effects of the higher sidelobes and broadening of the main beam. When the antenna dimension decreases the resolution decreases. To increase the resolution with the same elements (and thus the same gain), the aperture size must be increased, resulting in a sparse array. When the resulting degraded antenna pattern can be reconstructed, only the advantage of a better HPBW using a sparse array remains.

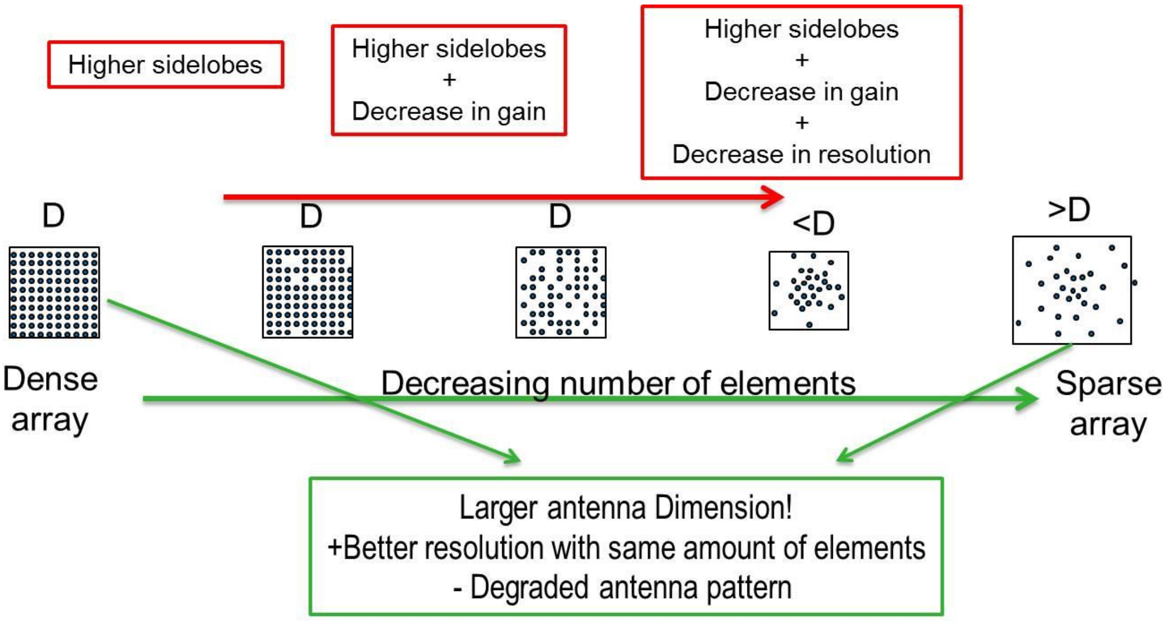
Higher sidelobes
Higher sidelobes + Decrease in gain
Higher sidelobes + Decrease in gain + Decrease in resolution
D
D
D
<D
>D
Dense array
Decreasing number of elements
Sparse array
Larger antenna Dimension!
+Better resolution with same amount of elements
- Degraded antenna pattern

Figure 10: Schematic overview of the advantages and disadvantages of using sparse arrays

3 Antenna Pattern reconstruction using Compressive Sensing

3.1 Introduction

Compression of data is well known for many years. Digital images are for example stored in "jpeg" format which can compress images up to one tenth of its original size. In the same way audio files are compressed using the MP3 format, discarding again about 90 percent of the data [5]. All these compression techniques have one important thing in common: they need the raw complete data before they can compress it. This is supported by the Nyquist-Shannon sampling criterion. If you want to fully recover your signal or data, you must at least sample at twice the highest frequency used. One could say this is full (uncompressed) sensing. Taking less samples results in recovery errors such as missing features or aliasing artifacts. This is also seen in our DOA estimation problem: when the element spacing in our array becomes larger than $\frac{\lambda_w}{2}$ grating lobes (aliasing) occur. Also when the array becomes sparse (taking less measurements) the antenna pattern gets distorted which results in a bad DOA estimation and possible missed targets. This in fact means that "original" information is discarded. This should however not always be a problem as can be seen in the compression techniques such as jpeg, MP3 and MPEG.

A good strategy for compression is to look for a domain in which the signal has a sparse representation. From this sparse domain then one could store only the *M* most prominent features instead of the total *N* measurements. By only storing these most prominent features and discard the rest a great reduction of data can be achieved without decreasing the original quality significantly. Taking all *N* measurements is however inevitable when one wants to apply compression. In our DOA estimation problem with a sparse array we only capture *M* features (in a sparse domain), without being sure these are the prominent ones needed to reconstruct the original signal to a satisfying level. When only the *M* most prominent features are captured this is called compressive sensing (CS), instead of compression.

CS theory says that certain signals can be recovered from far fewer measurements than traditional methods use. CS starts by looking for a domain in which the signal is represented in a sparse way and then only measure and save the most prominent features of the signal. The challenge in this is to find and measure only these *M* most prominent features without having to take all the *N* measurements. In this specific application of DOA estimation, the domain in which the signal is sparse is the angular domain. The sparsity of the array corresponds to the compressive sampling (in space).

An approach proposed in [6] by Emmanuel Candès, Justin Romberg and Terrence Tao is to look at it as an optimization problem. This approach aims to find the solution which is the most sparse i.e. the one that uses the least non-zero or most prominent measurements. Together with the work of David Donoho [7] it is the basic theory behind CS which is used in this thesis.

It is important to realize that the application of CS to Radar can be in the temporal- or spatial domain. The aim of using CS in the temporal domain is to reduce the number of samples needed for satisfying reconstruction of the original signal. This can be used to reduce the data stream since fewer samples than obligated by the Nyquist-Shannon criteria are needed. This requires a SNR > 0dB for CS to work. This is not obvious in radar since small targets have an SNR < 0dB at the receivers. Only after pulse compression and Doppler processing the SNR might be at an acceptable level for CS to work.

Obviously for this thesis the goal is to reduce the number of antenna elements and still be able to perform direction of arrival (DOA) estimation on a satisfying level. This means a reduction of the number of samples in the spatial domain instead of in the temporal domain. Most CS literature focusses on implementation in the temporal domain, for example in [8], [9], [10] and [11]. In [12], [13], [14] and [15] examples of where CS is applied in the spatial domain can be found. The results seem promising but in none of the papers enough information is given to easily reproduce the results. Also the effects of discretization errors and the accuracy and resolution of the solutions are not extensively addressed in these papers. Hence the theory is explained and implemented in Matlab to obtain results and insight in the effects of discretization errors and the accuracy and resolution when CS is used to reconstruct the signal of a sparse array.

In this chapter compressive sensing will be proposed as a technique to support signal reconstruction of sparse arrays for DOA estimation. First, in section 3.2, the initial CS problem with its principles will be formulated. The conditions needed for the principle of CS to work will be explained in this section, and also directly related to the DOA estimation problem using the ULA from chapter 2. Secondly, in section 3.3, there is a short discussion on discretization errors. In section 3.4 the signal reconstruction will be chosen and the implementation in Matlab will be explained. The fifth section concludes the chapter with the simulation setup used to obtain the results.

3.2 Basics of Compressive Sensing

The CS framework is described in [7], [13], [16] and [17]. In this section the important parts in relation to DOA estimation are addressed. Although some similarities with the DOA estimation problem are instantly clear others might be less obvious. Therefore the CS framework is formulated and explained. After that the conditions coming forth out of the formulation will be directly related to the DOA estimation problem using a ULA.

3.2.1 Initial problem formulation

Suppose that a (complex) signal is represented in a [*N* x 1] column vector **x** ϵ $\mathbb{C}^N$. Now let this signal be written as **x** =**Ψz**, where **Ψ** is the [*N* x *N*] sparsity basis matrix and **z** an [*N* x 1] K-sparse source vector. *K*-sparse means that only *K*<*N* entries in the vector are non-zero and large enough. The goal is to reconstruct the [*N* x 1] signal **x** using [*M* x 1] measurements, with *M*<*N*, which are represented in the measurement vector **y**. Normally when *M*>*N* the system is over determined and a solution can be found with, for instance, the use of the least squares method. Since *M*<*N* this means that the system is underdetermined, which makes it not possible to find a unique solution with conventional methods. Now "the CS theory states that **x** can be reconstructed using $M \cong K\log(N)$ non-adaptive linear projection measurements on to an [*M* x *N*] sensing matrix **Φ** that is incoherent with **Ψ**" [13]. Using the notation with sparsity basis matrix **Ψ** and the sensing matrix **Φ,** the measurement vector **y** can be written as:

$$\boldsymbol{y} = \boldsymbol{\Phi x} = \boldsymbol{\Phi\Psi z} = \boldsymbol{\Theta z}. \tag{16}$$

where **Θ** is an [*M* x *N*] observation matrix. The measurement process is non-adaptive which means that the sensing matrix **Φ** is fixed and does not depend on the original signal **x**. It is important to design a stable measurement matrix **Φ**, such that no information in the *K*-sparse source signal **z** is lost by the reduction in dimension from *N* to *M*. When this is guaranteed a reconstruction algorithm must be designed that recovers the *N*-length signal **x** from the *M*-length signal **y**, with *M*<*N*. These two problems make it that compressive sensing relies on two principles: incoherence and sparsity. They will be addressed in the next sections.

3.2.2 Design a stable sensing matrix

The sensing matrix $\mathbf{\Phi}$ reduces the *N*-length signal **x** to an *M*-length signal **y**. Since *M* < *N* this makes the problem underdetermined. If however **x** is *K*-sparse and the *K* locations of the nonzero coefficients in **z** are known, the problem can be solved under the condition that you at least “measure” every *K* sparse locations in **z**, e.g. *M* ≥ *K*. Of course in general the locations of the *K* non-zero components in **z** are not known and more measurements are needed to guarantee that no information is lost by applying the sensing matrix. In [17] the relationship between the number of *K* sparse components and the number of random measurements *M* needed for “a solution which is exact with overwhelming probability” is:

$$M \ \geq C\ \mu^2(\mathbf{\Phi},\mathbf{\Psi})K\log(N) \tag{17}$$

with C “some positive constant close to one” and $\mu(\mathbf{\Phi},\mathbf{\Psi})$ the coherence between $\mathbf{\Phi}$ and $\mathbf{\Psi}$ [17]. The coherence between $\mathbf{\Phi}$ and $\mathbf{\Psi}$ is given by:

$$\mu(\mathbf{\Phi},\mathbf{\Psi}) = \ \sqrt{N}\ \max_{1\leq i,j\leq N}\left|\varphi_{i,}\psi_j\right|. \tag{18}$$

Meaning that the coherence is the largest correlation between any two elements of $\mathbf{\Phi}$ and $\mathbf{\Psi}$**.** If $\mathbf{\Phi}$ and $\mathbf{\Psi}$ contain correlated elements, the coherence is large. Hence the reconstruction using CS relies not only on the sparsity *K* of the signal but also on coherence of the sensing matrix $\mathbf{\Phi}$ with the sparsity basis matrix $\mathbf{\Psi}$. The role of the coherence is clear: the lower het coherence between $\mathbf{\Phi}$ and $\mathbf{\Psi}$, the fewer measurements *M* are needed for reconstruction with “overwhelming probability” [17].

A low coherence between two matrices basically means that taking all subsets of *K* columns taken from the observation matrix $\mathbf{\Theta}$ are nearly orthogonal. Hence we would like to find observation matrices which can have as large subsets as possible while these are still nearly orthogonal. The larger the subsets the closer we come to the situation *M*=*K*, the minimum number of measurements needed if the sparsity profile was known. This condition was introduced by Candes and Tao in [18] and named the restricted isometry property (RIP). The RIP characterizes matrices which are nearly orthonormal, at least when operating on sparse vectors. It is difficult to proof the RIP for a concrete matrix, but Candes and Tao have proven that both the RIP and low coherence can be achieved with overwhelming probability by selecting the observation matrix $\mathbf{\Theta}$ as a random matrix provided that:

$$M \ \geq C\ K\log\left(\frac{N}{K}\right), \tag{19}$$

with *C* some constant (close to one) depending on each instance. The probability of sampling a matrix not obeying the RIP when (19) holds is “exponentially small in *M*” [17]. Finally it is stated in [17] that the RIP can also hold for the observation matrix $\mathbf{\Theta} = \mathbf{\Phi\Psi}$. If $\mathbf{\Psi}$ is a fixed arbitrary orthobasis and $\mathbf{\Phi}$ is an [*M* x *N*] sensing matrix drawn random from a suitable distribution, then the matrix $\mathbf{\Theta} = \mathbf{\Phi\Psi}$ obeys the RIP provided that equation (19) is satisfied. This is shown in Figure 11: although the sparsity basis matrix is a fixed Fourier spectrum matrix, after applying the random sensing matrix the result is another random observation matrix $\mathbf{\Theta}$. Due to this the sensing matrix can also be called universal: there is no need to know the sparsity basis matrix when designing the random sensing matrix $\mathbf{\Phi}$.

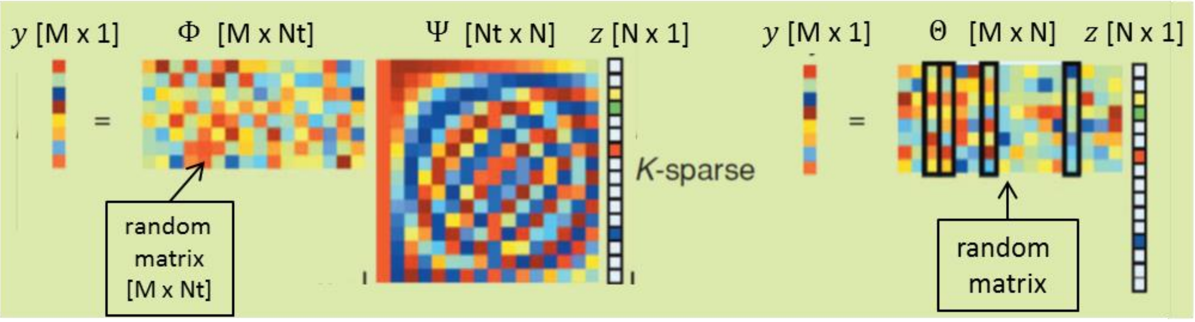

Figure 11: Compressive sensing measurement process with a random sensing matrix Φ and a fixed (Fourier spectrum) sparsity basis matrix Ψ.

In our antenna array the sensing matrix $\mathbf{\Phi}$ results from the fact that a dense array becomes sparse. The signal of a dense array is given by equation (12) as $\mathbf{x} = \mathbf{\Psi s} + \mathbf{w}$. Now let *N*-*M* out of *N* elements fail in the array, then the measurement vector $\mathbf{y}(t)$ can be described as:

$$\mathbf{y} = \mathbf{\Phi x} \tag{20}$$

With $\mathbf{\Phi}$ an [*M* x *N*] identity matrix with the rows corresponding to the failed elements removed as shown in Figure 12. Note that $\mathbf{\Psi}$ is still the scan angle matrix of equation (10), defining the discretization grid.

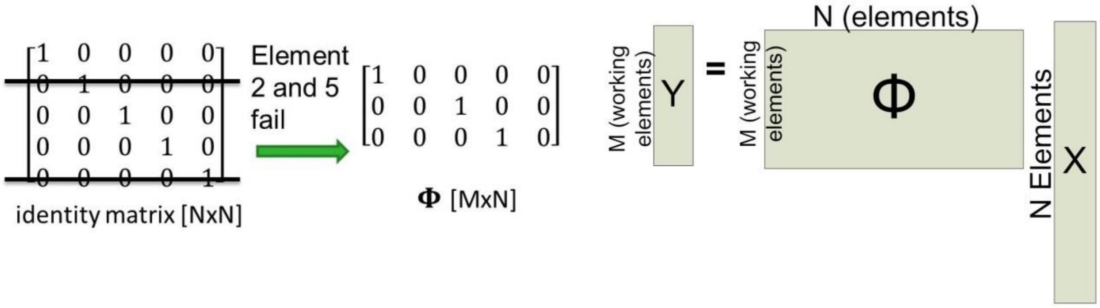

Figure 12: Creating the measurement matrix Φ (left). Dimensions of the measurement vector y, the measurement matrix Φ, and the original signal vector x (right).

The goal is to use compressive sensing techniques to reconstruct the signal of a sparse array as if it was a fully dense, $\frac{\lambda_w}{2}$ spaced ULA. Using equation (12) this results in the measured signal of the sparse array:

$$\mathbf{y} = \mathbf{\Phi x} = \mathbf{\Phi \Psi s} + \mathbf{\Phi w} = \mathbf{\Theta s} + \mathbf{\Phi w} \tag{21}$$

So far this perfectly suites our DOA estimation problem with sparse arrays. The random sensing matrix $\mathbf{\Phi}$ obeys the RIP as long as the minimal number of measurements given by equation (19) is satisfied. This means also the observation matrix $\mathbf{\Theta}$, obeys the RIP since it is the result of applying the random sensing matrix $\mathbf{\Phi}$ on the fixed basis of the scan angle matrix $\mathbf{\Psi}$.

Finding **s** from the *M* measurements of **y** is the task in CS. The signal $\mathbf{x} = \mathbf{\Psi s}$ is the reconstruction of the signals that would have been received by the dense array. Normally in CS one looks for the fewest number of measurements *M* needed to reconstruct the original *N* valued signal. In our DOA estimation problem however the original signal comes from *N* elements in a dense $\frac{\lambda_w}{2}$ spaced ULA. And the fewest number of measurements *M* are the remaining elements in the sparse array which is created by removing *N-M* random elements. This means in practice that the number of measurements *M* is a fixed number which cannot be changed easily once the array is built. Hence the number of remaining elements in the sparse array define the number of *K* targets which can maximally be reconstructed with “overwhelming” probability. Using (17) this results in:

$$K \leq \frac{M}{\log N} \tag{22}$$

targets which can be recovered when a sparse array is used with *M* instead of *N* elements in a fully dense array.

3.2.3 Designing a signal reconstruction algorithm

To reconstruct the vector **z**, and hence **x**, it is important that, besides the conditions regarding the incoherence between the basis matrix $\mathbf{\Psi}$ and the sensing matrix $\mathbf{\Phi}$ are met, also the sparsity criteria are met. The sparsity criteria are as follows:

a) In the first place the signal **x**, must have a sparse representation with respect to the sparsity basis matrix $\mathbf{\Psi}$. This means that the vector **z** is *K*-sparse, having K non-zero entries which are large and *N-K* entries which are (close to) zero.

b) Looking ahead to our DOA problem, we can say that there is sparsity in our solution since we expect only reflections from targets, and not from the surrounding air. Generally this will be only a few reflections in the total angular domain, making the source signal sparse in this angular domain. (It is thus also expected that for instance small targets in the presence of clutter cannot be reconstructed).

When the conditions for the incoherence and sparsity are met, the reconstructed source vector $\hat{\mathbf{z}}$ can be found by solving the following optimization problem:

$$\hat{\mathbf{z}} = \arg\min_{\mathrm{z}} \|\mathrm{z}\|_{\mathbf{1}} \;\; s.t.\, y = \mathbf{\Theta z}. \tag{23}$$

Where $\|\cdot\|_{\mathbf{1}}$ stands for the ℓ_1 norm. The p-norm of a vector is defined as:

$$\|x\|_p = \sqrt[p]{\sum |x_i|^p} \tag{24}$$

In our application we are not dealing with the sparse vector **z**, but with the source vector **s**. The prerequisite for the reconstruction algorithm to work is thus that **s** must be sparse. When there are not too many incoming targets (and the sidelobes are not too high), **s** is basically a sparse vector. In the columns corresponding to the angles of arrival there are high non-zero values. In the other columns there are low values, since they correspond to angles at which no target is present, and thus no reflections are received. Due to this it is expected that, using CS, in principle a sparse representation **z** of the source vector **s** can be found, provided that there are only *K* targets present.

3.2.4 Why the ℓ_1 norm is used

A well-known method using the norm is the method when p=2. This is the least squares or ℓ_2 norm, which calls for minimizing the sum of the squares of the measurements. Algorithms to find the solution by the least squares method are efficient, however the ℓ_2 norm tends to spread the energy, returning a non-sparse vector with many non-zero elements. Hence in compressive sensing, which looks for a sparse solution, the least squares solution is seldom the correct one [5].

A better way is to look for the solution with the least non-zero entries, e.g. the ℓ_0 norm. Modifying the optimization problem (23) with the ℓ_0 norm using p=0 in (24) results in exact recovery with high probability using only K+1 measurements. Unfortunately solving the ℓ_0 norm optimization problem is hard to compute because it is NP-hard and numerically unstable, requiring an exhaustive enumeration of all $\binom{N}{K}$ possible locations of the nonzero entries in **z** [16].

A compromise is to split the difference between "finding often wrong solutions" with the ℓ_2 norm and "hard to compute but correct" ℓ_0 norm solutions, is optimization based on the ℓ_1 norm. The ℓ_1 norm finds a solution with the minimum for the sum of all measured values. The difference between solving the optimization problem (23) with the ℓ_1 norm and the ℓ_2 norm is shown in Figure 13. A real valued sparse signal of around 500 values is reconstructed using only 60 complex valued Fourier coefficients. The task is to reconstruct this signal, while keeping the entries that should be zero really low. Figure 13(b) shows the ℓ_1 norm minimization and Figure 13(c) the ℓ_2 norm minimization. It is clear that the ℓ_2 norm minimization results in a minimum energy solution which does not provide a reasonable approximation to the original sparse signal. The ℓ_1 norm minimization however gives an exact reconstruction of the sparse signal.

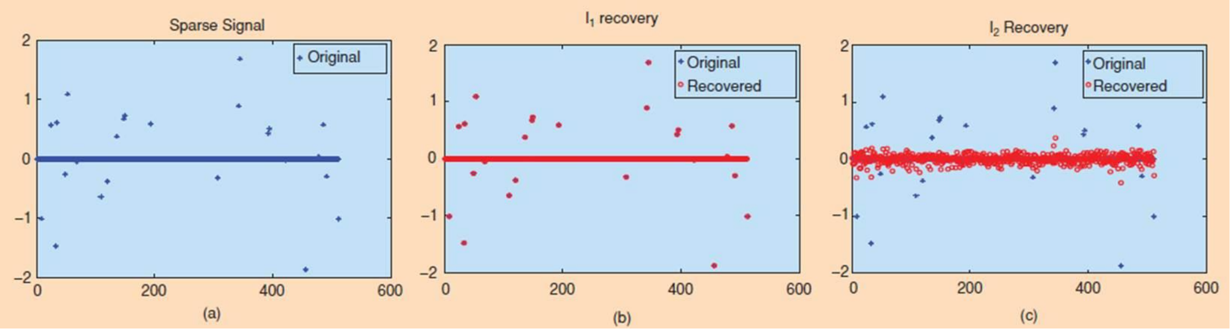

Figure 13: A sparse signal (a) and its reconstruction from 60 complex valued Fourier coefficients using the ℓ_1 norm minimization (b). The reconstruction in (c) is obtained by replacing the ℓ_1 norm with the $\ell_2$norm minimization. From [17].

Now the solution to our sparse vector **z** can be approximated by solving (23), the only remaining question is how to solve this optimization problem? Various techniques from the field of computer science can be used to solve this optimization problem. Since the emerging of the field of compressive sensing many algorithms based on various techniques are developed and available [19] [20]. In section 3.4 of this thesis a suitable algorithm is chosen and explained.

3.2.5 Using multiple time snapshots

Now the DOA estimation problem is formulated as a CS problem it can be solved using CS recovery algorithms. However in practical applications multiple time snapshots can be available. It is advantageous to use multiple time snapshots to reconstruct the signal because the solutions have a stronger uniqueness than when a single time snapshot is used [21]. This means that for a K-sparse signal the number of possible sparse solutions, that fit the limited number of measurements, is decreased as more time snapshots are used to reconstruct the original signal. A stronger uniqueness result gives thus less probability that non-unique and wrong reconstructions are the outcome of the reconstruction algorithm. This again can result in an increase of the number of targets that can be recovered, since for a single time snapshot with an increasing *K* also the probability of multiple (non-unique) solutions increases. Another advantage of using multiple time snapshots is that results are far less sensitive to noise [22]. The effect noise may also weaken the better uniqueness results but to what extent depends on the SNR.

Another question that arises is how multiple time shots can be used for the reconstruction in a radar system. The use of multiple time snapshots, if possible, must be applied in the right stage of the radar processing chain. In radar there are several processing stages which can influence the performance of the reconstruction with CS. Since sparsity is an important factor for CS, in thesis it is assumed that pulse compression is already applied in order to reach a sufficient SNR and "sparsity in range". Another important processing step is Doppler processing in which the Doppler effect is used to create Doppler bins in which targets with different radial velocities are presented. In this thesis the targets are assumed stationary, resulting in no radial velocity and a Doppler frequency equal to 0. When the targets do have different angular velocities, it could be advantageous to do the Doppler processing step before the reconstruction step. When the targets with different angular velocities are placed into different Doppler bins this creates an additional level of sparsity and a higher SNR. Studying the place and (dis)advantages of the reconstruction in the processing chain is beyond the scope of this thesis and recommended for further research.

To exploit the use of multiple time snapshots the framework must be adapted properly before we start working with the algorithm [13]. Start with rewriting the signal model of equation (12) to group *T* snapshots as:

$$\mathbf{X} = \mathbf{AS} + \mathbf{W} \tag{25}$$

where $\mathbf{X} = [\mathbf{x}(1)\ \mathbf{x}(2) \dots \mathbf{x}(T)]$ is a matrix of size [*N x T*], **A** is a matrix of size [*N x K*] as in (9), $\mathbf{S} = [\mathbf{s}(1)\ \mathbf{s}(2) \dots \mathbf{s}(T)]$ is a matrix of size [*K x T*] representing the sources, and $\mathbf{W} = [\mathbf{w}(1)\ \mathbf{w}(2) \dots \mathbf{w}(T)]$ is a matrix of size [*N x T*] representing the noise. Now define $\theta_1, \theta_2, \dots, \theta_{\mathrm{Ns}}$ be a set of angles with N_S the total number of angles we want to scan. The set of N_S angles thus defines our scan grid or [*N x N_s*] scan angle matrix $\mathbf{\Psi}$ as in (10). Next define an [N_S *x 1*] sparse vector **z** as:

$$\mathbf{z}(\mathrm{t}) = \left[z_{\theta_1}(\mathrm{t}) z_{\theta_2}(\mathrm{t}) \dots z_{\theta_{N_S}}(\mathrm{t})\right]^T, \tag{26}$$

with *K* nonzero values $z_\theta(\mathrm{t})$ at positions corresponding to the source angles $\theta = \theta_k, k = 1, 2, \dots, K$ and zero values at the remaining N_s-*K* positions. Now, using the scan angle matrix $\mathbf{\Psi}$ of (10), the signal model of (25) can be rewritten as a CS problem

$$\mathbf{X} = \mathbf{\Psi Z} + \mathbf{W}, \tag{27}$$

where $\mathbf{Z} = [\mathbf{z}(1)\ \mathbf{z}(2) \dots \mathbf{z}(T)]$ is a matrix of [N_S *x T*], hence containing a value for the received reflected energy from a target at scan angle θ at time snapshot *T*. It is assumed that $\mathbf{z}(1)\ \mathbf{z}(2) \dots \mathbf{z}(T)$ are jointly sparse or has a common sparsity profile. This means the indices of the nonzero entries are independent of *T*. In practice this is true as long as the angular velocity of the target is low. More general the assumption of a common sparsity profile means that the number of snapshots *T* is smaller than the number of sensors *M*, hence for this case *T* < *M*.

Assuming **Z** has a common sparsity profile, the angle scanning matrix $\boldsymbol{\Psi}$ is therefore still a sparse basis of the received array signals. The compressed array signal of (27) can now be rewritten as:

$$\mathbf{Y} = \boldsymbol{\Phi}\mathbf{X} = \boldsymbol{\Phi}\boldsymbol{\Psi}\mathbf{Z} + \boldsymbol{\Phi}\mathbf{W}, \tag{28}$$

Where $\mathbf{Y} = [\mathbf{y}(1)\ \mathbf{y}(2) \ldots \mathbf{y}(T)]$, being a [*M x T*] matrix containing the received signals per element at time snapshot *T*. Let $\widehat{\boldsymbol{Z}} = [\hat{\mathbf{z}}(1)\ \hat{\mathbf{z}}(2) \ldots \hat{\mathbf{z}}(T)]$ be the recovered solution, then the angle spectrum can be calculated by:

$$\mathbf{P}_{\mathbf{y}}(\theta) = \frac{1}{T}\sum_{t=1}^{t=T} \|\hat{z}_{\theta}(t)\|^2 , \theta = \theta_1, \theta_2, \ldots, \theta_{\mathrm{Ns}} \tag{29}$$

Since $\widehat{\mathbf{Z}}$ is a truly sparse vector, it just gives a peak at the exact DOA in the angle spectrum. An example, in which $\mathbf{P}_{\mathbf{y}}$ is plotted using the found $\widehat{\mathbf{Z}}$ obtained by the MFOCUSS algorithm (see section 3.4), is shown in Figure 14. Remember that the MFOUCSS solution is found using the signal received by the "defective" 16 element sparse array which has a scanned beam pattern with very high sidelobes!

Since $\widehat{\mathbf{Z}}$ is the found solution for the source matrix **S,** from equation (27) it becomes clear that the reconstructed signal $\widehat{\mathbf{X}} = \boldsymbol{\Psi}\widehat{\mathbf{Z}}$**.** The reconstructed signal $\widehat{\mathbf{X}}$ is used to determine the mean square error, as explained in 3.4.5.

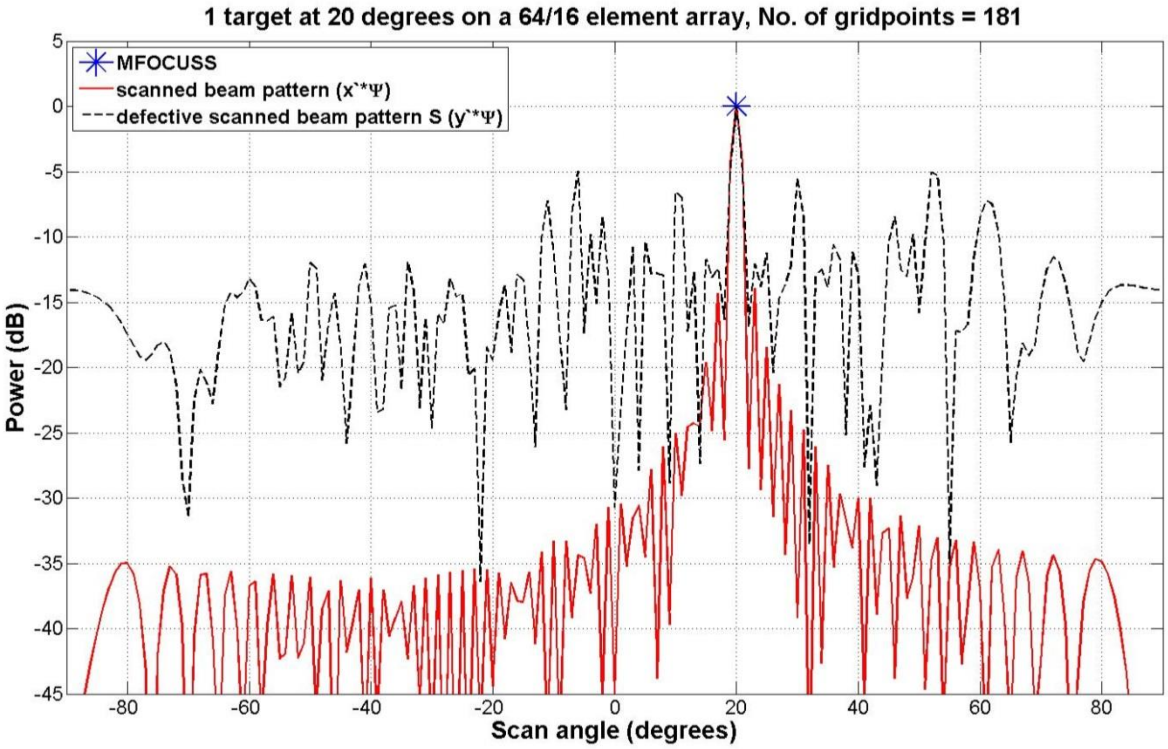

Figure 14: MFOCUSS solution compared to the scanned beam patterns of a fully dense 64 element array and a sparse array with 16 remaining elements. The antenna dimension is equal for both arrays.

Now the framework is defined and it is clear that the DOA estimation problem using an array antenna can be fitted into this framework, a suitable algorithm to solve the minimization problem of (23) can be sought. This is discussed in the next section.

3.3 Discretization errors

The sparsity basis matrix $\mathbf{\Psi}$, needed to apply CS, discretizes the continuous parameter of DOA angles into a finite set of grid points representing the DOA angles. Using this discrete nature of CS we assume targets lie exactly on grid points of the discretization grid. In practice this is obviously almost never true, since the target can arrive from every angle, so also at intermediate angles. This will give rise to discretization errors. One could choose a finer angular discretization, which results in the real DOAs being closer to the grid points. However when the grid size is very small there will be large coherence between the columns of the sparsity basis matrix. This higher coherence degrades the reconstruction performance, even beyond the point as described in (19) since this high coherence eventually results in higher computation load and numerical instability. These effects make it doubtful if choosing a very fine grid is beneficial for solving discretization errors [23]. Since the discretization errors and effects on the computational load and numerical stability are different for every situation, simulations are needed to see the effects of discretization errors. With the obtained results one should be able to choose a suitable discretization grid.

3.4 The signal reconstruction algorithm

Nowadays many algorithms can be found which try to find a solution for the sparse minimization problem (23). Obviously it is preferable that the used recovery algorithm is as fast and accurate as possible since we must process the data real time eventually in a real life system. Besides that it is important that the algorithm can find a unique sparse solution, even in the presence of noise. Also if there are any parameters to choose or tune, their influence on the solution must be known. Thus the performance of the algorithm has to be evaluated for several situations with different SNRs. Preferably the algorithm must be able to work with multiple time snapshots, often called multiple measurement vectors (MMV). In literature many algorithms based on various techniques are proposed [20] [21] [24]:

- ℓ_1 minimization algorithms;
- Greedy algorithms;
- Complexity based algorithms;
- Iterative reweighted least squares algorithms;
- And many more…

From literature [21] [24] a suitable algorithm for the CS DOA estimation problem was selected for this thesis: the (regularized) Multiple measurement vectors Focal Underdetermined System Solver (MFOCUSS) algorithm. MFOCUSS is based on the iterative reweighted least squares technique and is an MMV-extension of the original FOCUSS algorithm [21]. FOCUSS was originally designed to obtain a sparse solution by successively solving least squares optimization problems and is widely used to deal with compressed sensing problems. The advantages of FOCUSS are its low computation and stable results; only a few iterations tend to be enough to achieve a rather good approximating solution [24]. To deal with noise cases a regularized version of MFOCUSS was developed in [21]. This regularized version of the (M)FOCUSS algorithm also gives good overall performance for different SNR levels [21].

The MFOCUSS algorithm will be explained in this section, and the implementation in Matlab is discussed.

3.4.1 Multiple Measurement Vectors Focal Underdetermined System Solver

To understand the algorithm it is good to start with the original (unregularized) FOCUSS algorithm [22], [25], [26], since MFOCUSS is only an extension of FOCUSS for the use of MMV.

Consider the problem of solving the underdetermined system of equations:

$$\boldsymbol{y} = \mathbf{Ax} + \mathbf{n} \in \mathbb{R}^{M} \tag{30}$$

with **y** a vector containing [*M* x 1] measurements, **A** a [*M* x *N*] matrix, **x** the true source vector containing [N x 1] elements and **n** the additive noise. Leaving out the noise for the moment, there are infinitely many solutions since *M*<*N*. The most widely used method to find a solution that is nearly sparse is the minimum norm or least squares solution:

$$\min\|\boldsymbol{x}\|_2^2, \qquad \text{subject to } \mathbf{Ax} = \mathbf{y} \tag{31}$$

Where || . ||$_2$ denotes the ℓ_2 norm or the least squares solution. The solution to this problem can be represented in a closed form:

$$\mathbf{x} = \mathbf{A}^{\dagger}\mathbf{y} \tag{32}$$

Where $\mathbf{A}^{\dagger} = \mathbf{A}^{\mathrm{H}}\,(\mathbf{AA}^{\mathrm{H}})^{-1}$ denotes the pseudo-inverse. The least squares solution however has the tendency to spread the energy rather than obtaining a sparse solution. Hence this is where the derivation to the FOCUSS algorithm starts.

The FOCUSS algorithm finds localized solutions by starting with a distributed estimate that can be readily computed, such as the minimum norm. Then, in a series of repetitive steps, FOCUSS recursively enhances the values of some of the initial solution elements, while decreasing the rest of the elements until they become zero. In the end, only a small number of winning elements remain non-zero, yielding the desired type of sparse solution.

Instead of finding a solution to the original problem of (31) the algorithm is looking for a weighted least squares solution by including a weighing matrix **W** [25]:

$$\mathbf{x} = \mathbf{Wq}$$

In which **x** is again the unknown signal vector, **W** is a [*N* x *N*] weighing matrix and **q** is obtained by solving the minimization problem:

$$\min\|\boldsymbol{q}\|_2^2, \qquad \text{subject to } \mathbf{AWq} = \mathbf{y} \tag{33}$$

This results an optimal solution given by:

$$\begin{gathered}\mathbf{q} = (\mathbf{AW})^{\dagger}\mathbf{y}\\ \mathbf{x} = \mathbf{Wq} = \mathbf{W}(\mathbf{AW})^{\dagger}\mathbf{y} = \mathbf{WW}^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\left(\mathbf{AWW}^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\right)^{-1}\mathbf{y}\end{gathered} \tag{34}$$

This is the starting point of the FOCUSS algorithm which basically consists of two parts:

1. Find a low resolution estimate of the sparse signal;

2. Prune this solution to a sparse solution.

Initial low resolution estimate

Since the algorithm start pruning with the initialization, an initialization as close to the true solution as possible should be used. Hence when a priori information is available this should be incorporated in the initial estimate to constrain the solution further [18] [19]. (In [19] for instance the a priori information of the number of incoming targets is used to create an initial estimate which can refine the solution to a higher degree of accuracy. However for this thesis it is not assumed that the number of targets is known a priori). Since entries that are zero at the start of the pruning process remain zero for all iterations, it is important to choose an initial estimate which has only non-zero components. Otherwise potentially important components can be lost [13] [19]. It is however important to determine a unique initial estimate. The minimum 2-norm estimate given by $\mathbf{x} = \mathbf{A}^{\dagger}\mathbf{y}$ is such a unique initial estimate. Simulations have shown that when the minimum 2 norm solution was used as a starting point convergence to a solution was fastest [15]. The minimum 2 norm solution seeks a solution for **x** such that **x** is minimized. This is not a sparse solution, which is a good thing since then there are no zero components which can cause potentially important components to get lost.

Pruning to a sparse solution

The pruning process is based on weighted least squares minimization of the dependent variable with the weights being a function of the preceding iterative solutions [22]. Now let the (*k*-1)-th iteration of the estimate of **x** be given by [25]:

$$\mathbf{x}_{k-1} = [\mathrm{x}_{k-1;1}, \mathrm{x}_{k-1;2}, \dots, \mathrm{x}_{k-1;N}]$$

Then the next iteration *k* composes the a new weighting matrix $\mathbf{W}_k$ using the previous solution, and continues with the steps to calculate a new solution $\mathbf{x}_\mathbf{k}$. This can be described by 3 steps:

Step 1, calculate new weighting matrix:

$$\mathbf{W}_k = \begin{pmatrix} |\mathrm{x}_{k-1;1}|^{\mathrm{p}} & 0 & \cdots & 0 \\ 0 & |\mathrm{x}_{k-1;2}|^{\mathrm{p}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & |\mathrm{x}_{k-1;N}|^{\mathrm{p}} \end{pmatrix} \quad (35)$$

Step 2, calculate the new q-vector: $\mathbf{q}_k = (\mathbf{A}\mathbf{W}_k)^{\dagger}\mathbf{y}$

Step 3, calculate the new **x**-vector: $\mathbf{x}_k = \mathbf{W}_k\mathbf{q}_k$

With $\frac{1}{2} \leq p \leq 1$ being a parameter to compromise between sparsity and convergence. Hence in the first step of every iteration the weighting matrix **W** is updated using the previous solution. In step 2 and 3 the weighted least squares solution is computed using **W** from step 1. In this way the weights applied to the columns play a role in determining the contribution of the columns to the final solution. A small weight usually results in a smaller contribution and vice versa. In this way the low resolution initial estimate is pruned to sparse signal representation. The pruning is done by scaling the entries of the current solution by those of the solutions of the previous iteration. These 3 steps are repeated until the termination criteria are met. The parameter δ specifies when the algorithm is terminated by a convergence criterion which must be satisfied:

$$\frac{\|\boldsymbol{x_k} - \boldsymbol{x_{k-1}}\|_2}{\|\boldsymbol{x_{k-1}}\|_2} < \delta \tag{36}$$

The default value for δ is 10^{-8}. The algorithm stops when the average deviation between two successive iterations is smaller than δ.

The choice of the parameter *p* is dictated by the speed of convergence and the sparsity of the solution generated. When *p*=2 is set, MFOCUSS generates the ℓ_2 norm solution. Values of *p* ≤ 1 give sparse solutions, hence for sparse reconstruction *p* lies in [0,1]. The order of convergence is given by (2-*p)* and thus the algorithm converges faster for smaller values of *p* but there is also a higher likelihood to get trapped in local minima. In practice the default value of *p* = 0.8 have been found to represent a good compromise between speed of convergence and quality of the generated sparse solution [21].

An example of the pruning process from the initial estimate to a sparse solution is shown in Figure 15 . It can clearly be seen that the initial estimate is a spread minimum 2 norm solution. Every iteration the high power components get amplified since the weighing matrix is created using the previous solution.

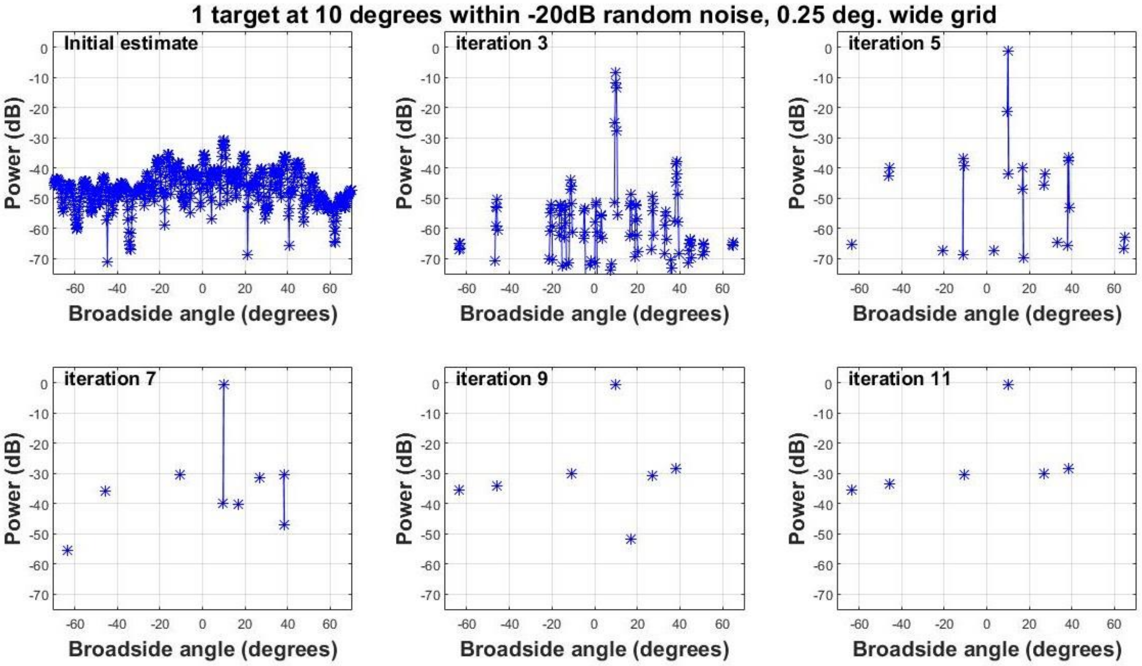

Figure 15: The Py vector for several iterations of the MFOCUSS algorithm, showing the pruning process.

3.4.2 Using Multiple measurement vectors

When multiple time snapshots or MMV are available, this can be exploited in order to increase the accuracy and uniqueness of the reconstruction of the signal (see section 3.2.5). It is noted that for the use of MMV it is assumed that the solutions have a common sparsity profile; i.e. the indices of the nonzero entries are independent of the obtained time samples. In this thesis stationary targets are assumed which gives no constraints for the use of MMV. In practice however, with a fast moving target, especially in the angular direction not to many samples can be used. On the other hand the difference in velocity of the targets might also be exploited trough Doppler filtering, as stated in section 3.2.5. Keeping in mind that in practice targets are not stationary, for this thesis the number of time samples/MMV that can be used is limited to ten.

When there are MMV available, equation (30) can be adjusted very easily by making the measurement vector **y** a measurement matrix **Y** which contains all the *T* measurement vectors. This also means that the noise vector **n** becomes a noise matrix **N** and obviously the source vector **x** becomes a source matrix **X**:

$$\mathbf{Y} = \mathbf{AX} + \mathbf{N} \in \mathbf{R}^{M} \text{ for some } M \text{ x } N \text{ measurement matrix } \mathbf{A} \tag{37}$$

3.4.3 Dealing with signals in noise

Now for the algorithm to deal with additive noise a regularized form of the algorithm was developed logically called regularized MFOCUSS [21]. The regularization is dictated by a parameter λ which determines the tradeoff between finding a solution as sparse as possible and with as small an error as possible. The regularization is applied in step 2 of the algorithm and is known as the Tikhonov regularization problem [27]:

$$\mathbf{Q}_{\mathrm{k}} = \arg\min\left(\|\mathbf{AW}_{\mathrm{k}}\mathbf{Q} - \mathbf{Y}\|_{\mathrm{F}}^{2} \quad + \quad \lambda\|\mathbf{Q}\|_{\mathrm{F}}^{2}\right), \tag{38}$$

where $\| \, . \, \|_{\mathrm{F}}$ denotes the Frobenius norm, which is explained directly after the steps of the regularized version of MFOCUSS. In equation (38) the part $\|\mathbf{AW}_{\mathrm{k}}\mathbf{Q} - \mathbf{Y}\|_{\mathrm{F}}^{2}$ is the part which controls the error of the solution, and $\lambda\|\mathbf{Q}\|_{\mathrm{F}}^{2}$ the part that controls the sparsity of the solution. Thus by choosing the regularization parameter λ one can make a trade-off between correctness and sparsity of the solution. (Note that setting the regularization parameter λ to zero, results in the original (non-regularized) MFOCUSS algorithm). The steps of the regularized MFOCUSS algorithm can be summarized as follows:

Step 1, calculate new weighting matrix:

$$\mathbf{W}_{k} = \begin{pmatrix} |\mathrm{x}_{k-1;1}|^{\mathrm{p}} & 0 & \cdots & 0 \\ 0 & |\mathrm{x}_{k-1;2}|^{\mathrm{p}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & |x_{k-1;N}|^{\mathrm{p}} \end{pmatrix} \tag{39}$$

Step 2, calculate the new **q**-vector: $\mathbf{Q}_{k} = \mathbf{A}_{k}^{\mathrm{H}}\left(\mathbf{A}_{k}\mathbf{A}_{k}^{\mathrm{H}} + \lambda\mathbf{I}\right)^{-1}\mathbf{Y} =$
Where $\mathbf{A}_{k} = \mathbf{AW}_{k}$ with $\lambda \geq 0$

Step 3, calculate the new **x**-vector: $\mathbf{X}_{k} = \mathbf{W}_{k}\mathbf{Q}_{k}$

As these 3 iteration steps from (39) are applied on equation (37) this now results in a obtained solution matrix **X**, instead of a solution vector **x.** Again these 3 steps are repeated until the termination criterion is met. The parameter δ specifies when the algorithm is terminated by a convergence criterion which must be satisfied:

$$\frac{\|\mathbf{X}_{k} - \mathbf{X}_{k-1}\|_{F}}{\|\mathbf{X}_{k-1}\|_{F}} < \delta \tag{40}$$

Where $\|\,.\,\|_F$ stands again for the Frobenius norm. The Frobenius norm is used to obtain the norm of a matrix instead of a vector. It is defined as the square root of the sum of the absolute squares of its elements e.g.:

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}\left|a_{ij}\right|^2} = \sqrt{Trace(\mathbf{A}\mathbf{A}^{\mathrm{H}})} \tag{41}$$

Where trace means to take the values on the diagonal of the matrix which is formed by multiplying **A** with its conjugate transpose. The MSE can also be calculated using the 2-norm (or Euclidian norm) instead of the Frobenius norm which is defined as:

$$\|\mathbf{A}\|_2 = \sqrt{\lambda_{max}(\mathbf{A}\mathbf{A}^{\mathrm{H}})} \tag{42}$$

Where $\lambda_{max}(\mathbf{A}\mathbf{A}^{\mathrm{H}})$ means to take the largest eigenvalue of $\mathbf{A}\mathbf{A}^{\mathrm{H}}$. Taking the square root of this largest eigenvalue is the same as taking the largest singular value. Since singular value decomposition is more computationally expensive than taking the Frobenius norm, hence usually the Frobenius norm is used [28].

When the matrix **A** becomes a vector, the Frobenius norm equals the 2-norm, since the 2-norm of a vector ***a*** is defined as $\|\mathbf{a}\|_2 = \sqrt{\sum_{i=1}^{n}|\mathbf{a}_i|^2}$. This equals *m*=1 in (41) and thus for a vector the Frobenius norm equals the 2-norm. Also for the initial estimate the minimum Frobenius norm estimate can be used instead of the minimum 2-norm that was used in the single measurement vector situation. See also section 3.4.5.

Finding the correct value for the regularization parameter λ, can be challenging. Several ways such as the modified L-curve method [21] and the generalized cross-validation (GCV) technique [29] are proposed to estimate the regularization parameter. In the MFOCUSS m-file [19], there is no method included to estimate the parameter λ and it has to be chosen by the user as one of the input parameters. In the m-file it is described as: “Regularization parameter, generally it is close to the noise variance. In the noiseless cases, simply setting lambda = 1e-10 leads to good performance.” With the latter being confirmed by equation (38), the challenge to estimate λ in noisy cases remains. Choosing a fixed value for λ can result in a suboptimal result since at every iteration the pruning process creates a new solution, which is used as an input for the next iteration. Hence the best trade off between sparsity and error, being λ, can easily be different for every iteration. As a result choosing a fixed value for λ can result in a lower rate of convergence, a larger error or a solution that is not sparse enough or even too sparse. This makes a way to determine λ at every iteration desirable.

In [21] and [27] the modified L-curve method is proposed as a method to find λ at each iteration. The modified L-curve method seeks the value of λ that minimizes both the error and the sparsity of the solution. This is done by determining the outcome of the error term $\|\mathbf{A}\mathbf{W}_{\mathrm{k}}\mathbf{Q} - \mathbf{Y}\|_F^2$ versus the outcome of the sparsity term $\lambda\|\mathbf{Q}\|_F^2$ for several values of λ. This plot results in an L-curve, with the distinct corner of the L being the optimal value for λ. This optimal value gives then the best tradeoff between sparsity and error at the current iteration. In this method it is however assumed that the boundary values of λ are known or can be estimated given a prior value of the SNR [29]. And experiments from [27] showed that not for every iteration the plot contains an L-curve.

In [29] the GCV is preferred over the modified L-curve method since it is more flexible for integration with the main algorithm. It is also easier to search for the minimum of the GCV function than to find the corner of the L-curve. Besides that, for the modified L-curve method, computation of the modified L-curve involves computation of the solution for several samples of λ. This results in an increased computational load. Using the GCV technique combined with a suitable minimization algorithm, λ can be numerically estimated with better computational efficiency and without solving the problem for several sample values of λ. Additionally the GCV usually gives a more accurate estimate of λ than the L-curve method [29].

The GCV technique estimates the optimal value for λ as follows. Given the M measurements (stored in **Y**), one measurement is left out and the value of this measurement is predicted by using the M-1 remaining measurements. The idea is that the best value for λ is the one that predicts each left out measurement the most accurate, as a function of the others. Therefore the estimate of λ is the minimizer of the function [30]:

$$P(\lambda) = \frac{1}{M}\sum_{m=1}^{M}([\mathbf{A}\mathrm{x}_m(\lambda)]_m - y_m)^2. \tag{43}$$

The GCV estimate is a variant of this form, obtained by applying singular value decomposition and some straightforward calculations, which results in the GCV function [29]. The GCV technique estimates λ by assuming that an optimum value of λ should be chosen to minimize the GCV function. The GCV function for the standard regularized MFOCUSS algorithm is derived in appendix II of [29] and defined as:

$$V(\lambda) = \frac{\sum_{i=1}^{M} \xi_{ii}\left(\frac{\lambda}{\sigma_i^2 + \lambda}\right)^2}{\frac{1}{M}\left(\sum_{i=1}^{M}\frac{\lambda}{\sigma_i^2 + \lambda}\right)^2} \tag{44}$$

Where the index ξ_{ii} = $\mathbf{U}^{\mathrm{H}}\mathbf{Y}\mathbf{Y}^{\mathrm{H}}\mathbf{U}\ \epsilon\mathbb{R}^{MxM}$, and **U** being a result of the singular value decomposition: $\mathbf{WA} = \mathbf{US}\boldsymbol{V}^{\mathrm{H}}$**.** To minimize $V(\lambda)$ the following constrained problem has to be solved:

$$\hat{\lambda} = arg \min_{\lambda min \leq \lambda \leq \lambda max} V(\lambda) \tag{45}$$

Equation (45) can be solved using the Matlab function “fminbnd” which is based on the golden section search and parabolic interpolation [29]. The function *“fminbnd”* finds the minimum of a function of one variable within a fixed interval. By choosing a sufficiently wide range for the boundary values λ_{min} and λ_{max} the function always finds a minimum. The estimation of λ using equation (45) must preferably be done at every new iteration of the MFOCUSS algorithm.

The advantage of the GCV technique is shown in Figure 16. Due to determining an optimal λ at every iteration the GCV solution (shown in red) converges faster and to a more sparse solution as the original regularized MFOCUSS solution with a fixed value of λ=0.01 (shown in blue), which is equal to the noise variance as stated in the description of the algorithm. It should be noted that the advantage of using the GCV technique to determine λ at every iteration is mainly useful for situations with discretization errors, e.g. targets being not on the grid. This is a situation which is expected to be very common in real life. When there are no discretization errors, using the same predefined λ at every iterations generally gives a faster convergence. For on grid situations the GCV technique has still two remaining advantages: reaching a more sparse solution without noise and not needing to define λ as an input parameter by the user, having an adaptive value for λ.

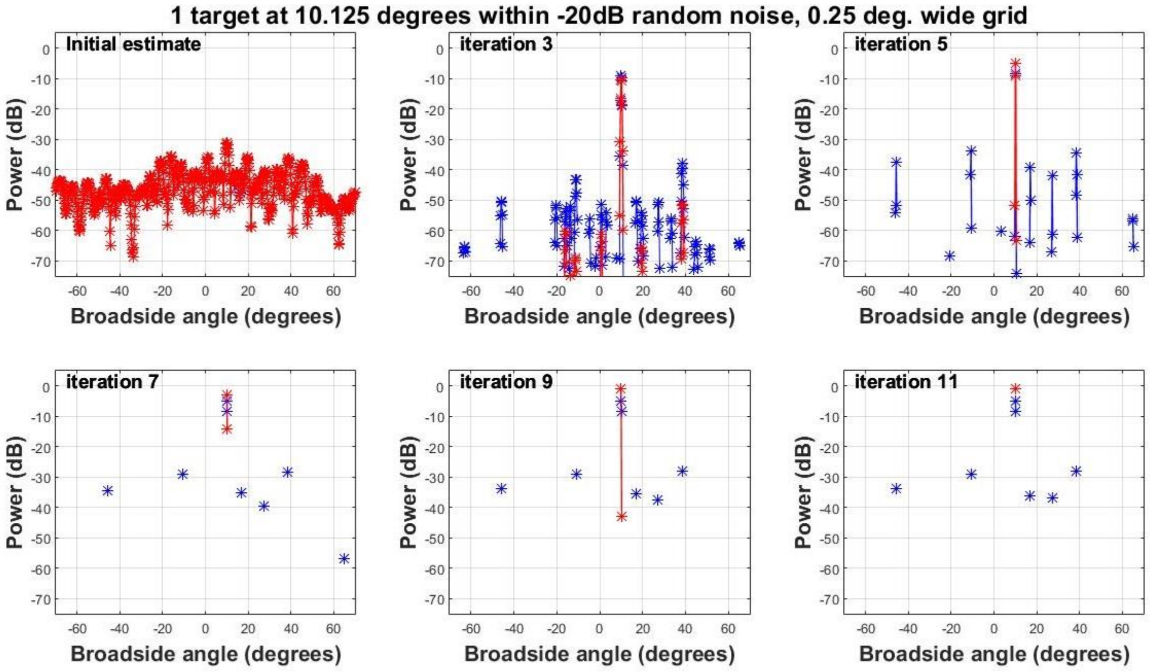

Figure 16: Pruning process for normal regularized MFOCUSS with λ=0.01 (blue) and GCV MFOCUSS (red).

3.4.4 Implementation of MFOCUSS in Matlab

In this section the implementation of the MFOCUSS algorithm into Matlab will be addressed. The algorithm is programmed in the m-file presented in Appendix A and is described as follows.

The MFOCUSS algorithm should be finding the (weighed) least squares solution to the problem (34):

$$\mathbf{Q} = (\mathbf{AW})^{\dagger}\mathbf{Y}$$

$$\mathbf{X} = \mathbf{WQ} = \mathbf{W}(\mathbf{AW})^{\dagger}\mathbf{Y} = \mathbf{WW}^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\left(\mathbf{AWW}^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\right)^{-1}\mathbf{Y} = \mathbf{W}(\mathbf{WA})^{\dagger}\mathbf{Y}$$

Where $(\mathbf{AW})^{\dagger}$denotes the pseudo inverse.

With **A** a [*M* x *N*] matrix representing the observation matrix, **W** a [*N* x *N*] weighing matrix and **Y** the [*M* x *T*] measurement matrix. The pseudo inverse is a generalization of the inverse matrix and therefore it is often used to find the minimum least squares solution. The computation of the pseudo inverse $(\mathbf{WA})^{\dagger}$ using the definition $\mathbf{W}^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\left(\mathbf{AWW}^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\right)^{-1}$ is computationally not advantageous for numerical reasons [26]. A better way to evaluate the pseudo inverse is to use and compute the singular value decomposition of **WA:**

$$\mathbf{WA} = \mathbf{US}\boldsymbol{V}^{\mathrm{H}} \tag{46}$$

where: **U** is an M × M orthogonal matrix: $\mathbf{U}^{\mathrm{H}}\mathbf{U} = \mathbf{I}$**,** with the columns containing the left singular vectors**;**
S is an M × N diagonal matrix containing the singular values σ_i of **WA;**

and **V** is an N × N orthogonal matrix: . $\mathbf{V}^{\mathrm{H}}\mathbf{V} = \mathbf{I}$**,** with the columns containing the right singular vectors.
Now the pseudo inverse is given by [26]:

$$\mathbf{WA}^{\dagger} = \mathbf{US}^{\dagger}\mathbf{V}^{\mathrm{H}} \tag{47}$$

For a rectangular diagonal matrix **S** the pseudo inverse can be obtained by replacing the singular values σ_i by their reciprocal $\frac{1}{\sigma_i}$, which is in fact the inverse **S**$^{-1}$. This makes a singular value decomposition a relative computationally efficient operation and suitable to use within the MFOCUSS algorithm.

For the regularization problem equation (38) must be solved which means that the singular value decomposition and finding the pseudo inverse must be applied on step 2 of equation (39), being:

$$\mathbf{Q_k} = \mathbf{A}_k^{\mathrm{H}}\left(\mathbf{A}_k\mathbf{A}_k^{\mathrm{H}} + \lambda\mathbf{I}\right)^{-1}\mathbf{Y}\text{, where } \mathbf{A}_k = \mathbf{AW}_k \text{ with } \lambda \geq 0$$

In this the pseudo inverse of $\mathbf{A}_k$ can be identified as below:

$$\mathbf{A}_k^{\dagger} = \mathbf{A}_k^{\mathrm{H}}\left(\mathbf{A}_k\mathbf{A}_k^{\mathrm{H}} + \lambda\mathbf{I}\right)^{-1}, \quad \text{where } \mathbf{A}_k = \mathbf{AW}_k \text{ with } \lambda \geq 0 \tag{48}$$

Using equation (47) and the fact that $\mathbf{A}_k = \mathbf{U}_k\mathbf{S}_k\mathbf{V}_k^{\mathrm{H}}$**,** with $\mathbf{U}_k^{\mathrm{H}}\mathbf{U}_k = \mathbf{I}$ and $\mathbf{V}_k^{\mathrm{H}}\mathbf{V}_k = \mathbf{I}$ finding the pseudoinverse results in:

$$\begin{aligned} \mathbf{A}_k^{\dagger} &= \mathbf{A}_k^{\mathrm{H}}\left(\mathbf{A}_k\mathbf{A}_k^{\mathrm{H}} + \lambda\mathbf{I}\right)^{-1} \\ &= (\mathbf{V}_k\mathbf{S}_k^{\mathrm{H}}\mathbf{U}_k^{\mathrm{H}}\ \mathbf{U}_k\mathbf{S}_k\mathbf{V}_k^{\mathrm{H}} + \lambda\mathbf{I})^{-1}\mathbf{V}_k\mathbf{S}_k^{\mathrm{H}}\mathbf{U}_k^{\mathrm{H}} \\ &= \mathbf{V}_k(\mathbf{S}_k^{\mathrm{H}}\mathbf{S}_k + \lambda\mathbf{I})^{-\mathbf{1}}\mathbf{S}_k^{\mathrm{H}}\mathbf{U}_k^{\mathrm{H}}\mathbf{.} \end{aligned} \tag{49}$$

Since the inverse of the diagonal matrix **S** and **I** is equal to its reciprocal this can be rewritten to:

$$\begin{aligned} \mathbf{A}_k^{\dagger} &= \mathbf{V}_k\ \mathrm{diag}(\frac{1}{\sigma_i^2} + \frac{1}{\lambda})\mathbf{S}_k^{\mathrm{H}}\mathbf{U}_k^{\mathrm{H}} \\ &= \mathbf{V}_k\ \mathrm{diag}(\tfrac{\sigma_i}{\sigma_i^2+\lambda})\mathbf{U}_k^{\mathrm{H}}\mathbf{,} \end{aligned} \tag{50}$$

with σ_i still being the singular values from **S**. Note that when $\lambda \rightarrow 0$**,** equation (50) is equal to equation (47), which is the problem without regularization. Equation (50) can properly be applied in Matlab since it only contains matrix operations. And hence the solution for the *k*-th iteration of the regularized MFOCUSS can be determined using:

$$\mathbf{X}_k = \mathbf{W}_k\mathbf{A}_k^{\dagger}\mathbf{Y} = \mathbf{W}_k\ \mathbf{V}_k\ \mathrm{diag}\left(\frac{\sigma_i}{\sigma_i^2 + \lambda}\right)\mathbf{U}_k^{\mathrm{H}}\mathbf{Y} \tag{51}$$

The only thing left is to implement a way to determine the regularization parameter λ at each iteration. As explained this can be done at each iteration using the GCV technique using equations (44) and (45). As stated in [29], λ can be found by minimizing equation (44) by solving the constrained problem of equation (45) using the function *"fminbnd"* from Matlab. The GCV function of equation (44) is written in Matlab and the constrained problem is solved using "*fminbnd"* on the fixed interval λ_{min}=10^{-10} and λ_{max}=10^{3}. This is done every time equation (50) is solved, thus every iteration the optimal λ is used and not predefining λ as in input parameter by the user is not necessary any more.

The implementation of GCV also increases the computational load. For extensive simulations this can result in a evidently larger computational burden since for every iteration the GCV procedure is carried out and the minimum of $V(\lambda)$ has to be evaluated. The effects of the width of the interval on which λ is evaluated is not clear when multiple targets are present. Since λ is a compromise between the sparsity and correctness of the solution, a large interval can result in erroneous solutions. A small interval on the other side takes a way a lot of the advantage of fast convergence and possibly finding a solution as sparse as possible. Studying these effects extensively is beyond the scope of this thesis and recommended for further research.

3.4.5 Performance parameters

Detection of targets is a very important issue in RADAR systems. CS techniques are relatively new and conventional detection methods cannot necessarily be applied directly on the CS output. Finding a suitable detection method for CS RADAR systems is a recent ongoing research area [5] [21]. The latter is beyond the scope of this thesis and the MSE and the method of "correctly identified columns" are used to measure performance since, for this thesis, we have knowledge of the true signals. When the target is off the grid, both the adjacent grid points are a correct solution.

The relative mean squared error (MSE) between the true and the estimated solution is used as a measure for the performance and in [21] is described as:

$$\mathrm{MSE} = \mathrm{E}\left(\frac{\|\widehat{\mathbf{X}} - \mathbf{X_0}\|_F^2}{\|\mathbf{X_0}\|_F^2}\right) \tag{52}$$

Where $\|\,.\,\|_F$ denotes the Frobenius norm, $\widehat{\mathbf{X}}$ is the recovered solution matrix and $\mathbf{X_0}$ is the true solution matrix. For this thesis the true solution $\mathbf{X_0}$ is known and used to create a sparse measurement vector and can hence be used to evaluate the obtained recovered solution using (52). To get an idea of what the MSE says about the found solution in Table 1 several values for the MSE, with different noise levels, are shown. The situation in which the MSE was obtained was an ideal case where an exact solution can be found. Hence purely the effect of the noise is shown in Table 1. It can be seen that the MSE is in the order of the noise variance.

Table 1: Values for the MSE for different noise values.

SNR [dB]	100	50	40	30	20	10	0
MSE	1.42e-10	1.42e-05	1.30e-04	1.30e-03	1.23e-02	0.1132	0.51

Since the interest lies in estimating the DOAs of the targets, another way of measuring the performance of the algorithm is simply to identify the correctly estimated DOAs. This can be done by looking for the (high) values in the $\mathbf{P_y}$ vector of equation (29) since they represent the reconstructed DOAs. The method that have been used is referred to as "correctly identified columns" [21] which looks if there is a non-zero value in the correct column of $\mathbf{P_y}$. It is noted that the "correctly identified columns" method only can work properly as long as the true number of targets and their DOAs are known. Thus it can only be used for testing purposes and not in real life.

3.5 Simulation setup in Matlab

For this thesis the CS theory is applied to a 64 element ULA with 48 random defective elements. The received signal vector **X** of the 64 element ULA is generated using Matlab. Then the signal vector **Y** of the sparse array is generated by multiplying **Y** with the random sensing matrix $\mathbf{\Phi}$. The only constraint on the sensing matrix is that row one and row 64 cannot be removed since this results in an equal antenna dimension *D* for the fully dense ULA and the sparse ULA. From the remaining 62 rows, 48 random rows are removed. Now the aim is to reconstruct the signal **X** and estimate the DOA of incoming targets correctly, using **Y**, $\mathbf{\Phi}$, a sparse basis and the MFOCUSS algorithm.

Unless stated otherwise the basic configuration that have been used is:

- A ULA with N=64 and M=16 random working elements, element 1 and 64 are always working;
- Element spacing for the fully working ULA: $\frac{\lambda_w}{2}$;
- Operating frequency 3 GHz;
- Working elements: 1, 10, 11, 14, 15, 23, 33, 37, 38, 41, 48, 49, 50, 59, 63, 64, see also Figure 17;
- 1 time sample used;
- Noise power: 1e-10 (-100dB) for the noiseless case, for the noisy case: as specified;
- Target power: 1 (0dB) ;
- Targets are stationary/have no angular velocity (Doppler frequency=0);
- The DOA of the targets is limited from -60 to 60 degrees (this is the practical scan range for a phased array, see also section 2.2.2 and 2.2.3)
- MFOCUSS parameters:
    - Regularization parameter λ = noise variance (unless stated otherwise);
    - *P* =0.8 (see section 3.4.1);
    - $\delta = 10^{-8}$ (default value)

Since the fully dense array consists out of 64 elements spaced $\frac{\lambda_w}{2}$, referring to equation (7) the beamwidth, of the fully dense array is approximately: $\theta_{3dB} = \frac{2}{N} \approx 1.79°$. The beamwidth of the sparse array is the same since the antenna dimension *D* is the same.

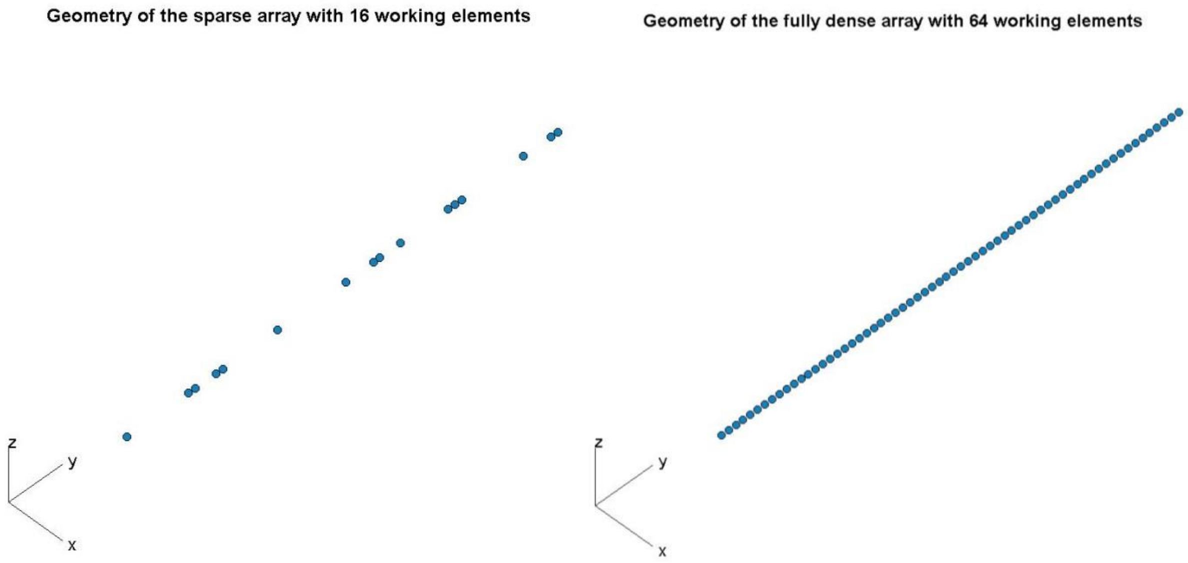
Geometry of the sparse array with 16 working elements
Geometry of the fully dense array with 64 working elements
z
y
x
z
y
x

Figure 17: Used array geometries

To keep results reproducible and consistent, one random configuration as specified above and in Figure 17 has been used. When random defects occur in real life this configuration used is obviously not always the result. To see if large deviations occur when other random configurations are used, simulations have been done using 10 different random arrays. The only constraint on these 10 random configurations was that element 1 and 64 are always working in order to keep the same antenna dimension *D* and hence same original resolution for the configurations. The results are summarized in appendix B, concluding that as long as the configuration is random the effect of different random arrays are comparable.

The solution angle grid is specified from -90 to 90 degrees off broadside, with a separation between the gridpoints of one degree. This result in using N_s=181 gridpoints. This makes the grid is finer than the beamwidth and hence the maximal achievable resolution (See also section 2.4 and [3]). Therefore the initial chosen gridsize of 1 degree makes it possible to achieve the maximal resolution and keep the computation load and possible numerical instability under control (see also section 3.3). During the research other gridsizes where used, if this applies this will be mentioned. One might argue that also the dimension of the grid can be reduced to the limited scan range and therewith reducing computational efforts. This is however a risky action since the grid specifies the scan angles on the receive side only. The transmit beam does not necessarily have very low side lobe levels. Hence signals may be received coming from the sidelobes of the transmit beam. Additionally in real life strong sources such as jammers can be coming from any direction, also from directions which are beyond the scan limit. These signals can influence targets in the main beam, and when the grid is limited the existence of these signals is not known. Truncate the grid at the sides to reduce computational efforts is thus not advisable.

As an example the resulting m-file for the simulation of *K* incoming targets, using the setup as described is included in appendix C.

4 Results

In this chapter results are presented and used to evaluate the use of CS, using MFOCUSS, for DOA estimation. The research approach is as follows:

1. Simulations have been done “without” noise (e.g. a noise level of -100dB) and targets exactly on the discretization grid. The number of incoming targets has been varied for these simulations. The results can be used to evaluate the performance of the unregularized MFOCUSS algorithm itself. An important issue to evaluate is whether the theoretical relation between the number of measurements and number of sparse components is satisfied.
2. The effects of discretization on an angular grid are examined, again for varying numbers of targets, have been examined without the presence of noise.
3. Noise has been added to the simulations, for varying number of targets, to examine the effect on the reconstructed solutions.
4. The effect of discretization errors on the angular grid has been evaluated also in the noise case to create a realistic scenario and to see the effects on the solutions.
5. The MMV configuration has been considered for the noisy case with off-grid targets.
6. Simulations have been done to obtain progressive insights in the effects of changing the regularization parameter λ.
7. Simulations have been done to determine the accuracy and resolution of DOA estimation using CS, and the effect of choosing a very fine angular discretization grid.

4.1 Recovery without noise

4.1.1 Single target on the grid

The simplest starting scenario is the arrival of a single target on grid without the presence of noise. Targets with different DOAs have been simulated and their angle spectrum according to (29) can be plotted. Three examples are shown in Figure 18, in which also the scanned beam pattern obtained with the fully working dense array is shown. As explained in section 2.3 the scanned beam pattern is calculated trough $\mathbf{X}^*\boldsymbol{\Psi}$. Because the vector $\mathbf{X}$ is used, this is the scanned beam pattern of the fully working dense array.

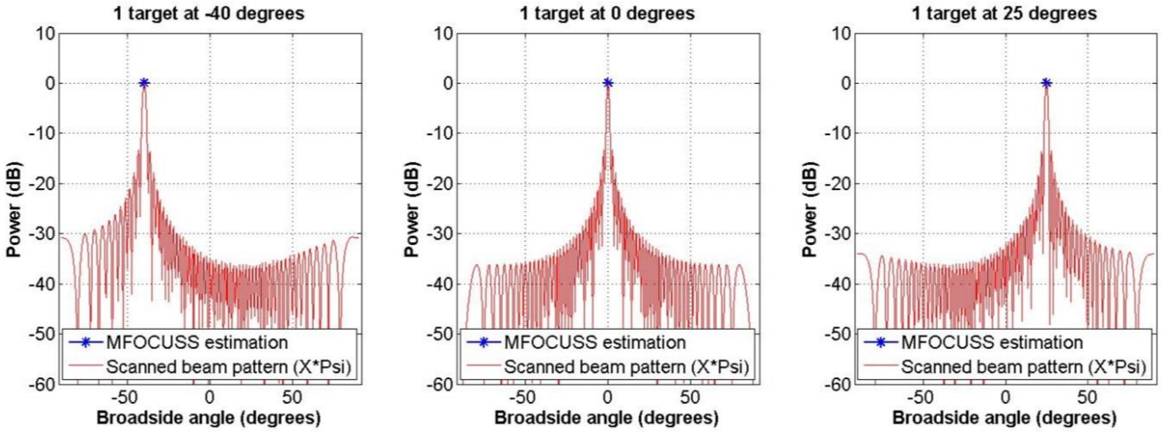

Figure 18: Reconstruction of a single target on the grid with no noise.

Since there is no noise and the target DOA is on the grid, only a single solution is found. The other solutions, if present are below -100dB since this is the noise level for the no noise situations. As explained in section 3.4.5 also the MSE can be used to measure the performance of the algorithm. The MSE is based on the mean squared difference between the reconstructed signal $\widehat{\mathbf{X}}$ and the true received signal **X**. In Table 2 the MSE and the number of iterations needed are shown for each DOA. In the case of on-grid targets, once the reconstruction is correct, the MSE is not meaningful, since the DOA is estimated exactly on the correct grid-cell. Therefore the MSE has to be zero unless of numerical precision error. This is evident from the graph in Figure 19, where the reconstructed and known angular responses for the target at 25 degrees from Figure 18 are identical. Since the interest lies in the DOA estimation, in general the angle spectrum according to (29) was used in figures and the MSE was calculated as a measurement of performance.

Table 2: MSE, number of iterations needed for a single on grid target without noise for different DOAs

DOA	-40	0	25
MSE	1.41e-10	1.42e-10	1.58e-10
Iterations	9	8	8

The number of iterations needed is maximal 15 for the DOA of-50 degrees, meaning convergence is reached within these 15 pruning steps. This is due to the fact that there is no noise and the DOA is on grid, which results in minimal deviations that can make the solution not exact and thus not have to be solved by the algorithm.

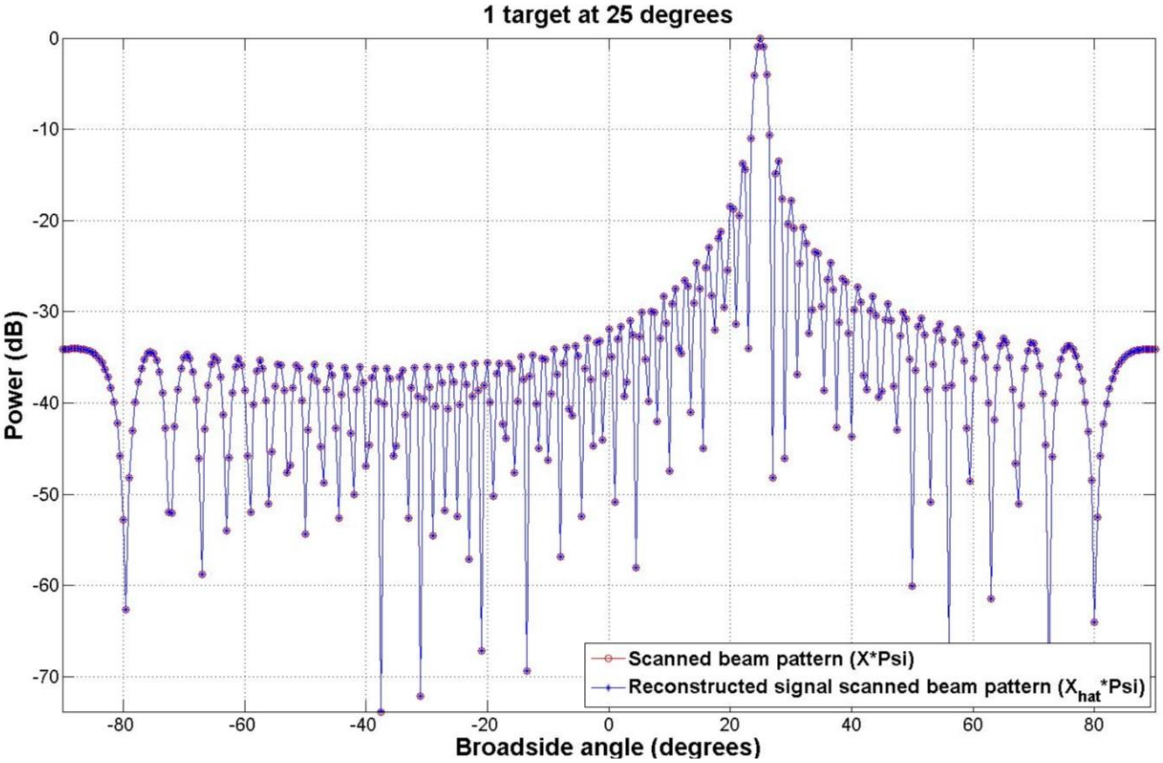

Figure 19: Scanned beam pattern and reconstructed scanned beam pattern for the reconstruction of a single target with a DOA at 25 degrees without noise.

4.1.2 Maximal amount of targets that can be solved

From section 3.2.2 and equation (17) the relation between the number of non-zero components of the signal and the number of needed measurements for recovery is known. Since the coherence is almost equal to 1 in the case of a DOA problem (because of the non-orthogonal closely spaced grid) at least K log (N) elements should be available in the sparse array to recover K targets. However the number of total elements N and number of remaining elements M are fixed at 64 and 16 respectively in this case. Thus it is expected from theory that there can be $M/\log(N)$ = **3.85** targets recovered, see section 3.2.2 and equation (22). The expectation is that when 4 or more targets are present at the same time the reconstructed solution will not be correct with “overwhelming probability”.

Equal strength targets

To verify this theoretical limitation, 1000 individual experiments have been done for a different number of K targets at random on grid DOAs between -60 and 60 degrees. The random DOAs have been created by picking K random integers between -60 and 60 in which doubles within the K random picked DOAs are avoided.

Table 3: Number of solutions with incorrect identified columns and average MSE for random DOA estimations of up to 6 targets.

No of targets	1	2	3	4	5	6
Wrong solutions within 1000 simulations	0	0	0	28	168	430
Average MSE	1.38E-10	7.40E-11	4.74E-11	0.0166	0.0934	0.254

In Table 3 the numbers of solutions with at least one incorrect identified column are summarized. As expected from theory, the errors occur when 4 or more targets are present. Since the theoretical limit of 3.85 targets that can be recovered with “overwhelming” probability holds, from 4 targets wrong reconstructions start to appear. An example of a wrong reconstruction with 4 targets is shown in Figure 20. It can be seen that the targets are not closely spaced but are separated at least 23 degrees. The failed reconstruction is purely related to the fact that the criterion (K≤3.85) for the available number of measurements (M=16) is not satisfied.

The average MSE in Table 3 is also increased significantly for K=4. From Table 1 it is known that when reconstructions can be exact, the MSE is in order of the noise variance if these solutions are exact. For the MSE of K<4 this is indeed true, but for K=4 the MSE is much higher than expectations based on the noise variance of -100dB. This is caused by the 28 wrong reconstructions only: the 972 correct reconstructions all have an MSE<10^{-10} since they are exact reconstructions. For the 28 wrong reconstructions 0.12 ≤ MSE ≤ 0.94 applies. Hence also the (average) MSE indicates that there are wrong reconstructions. This trend continues for K>4: both the MSE and the solutions with incorrectly identified columns increases .

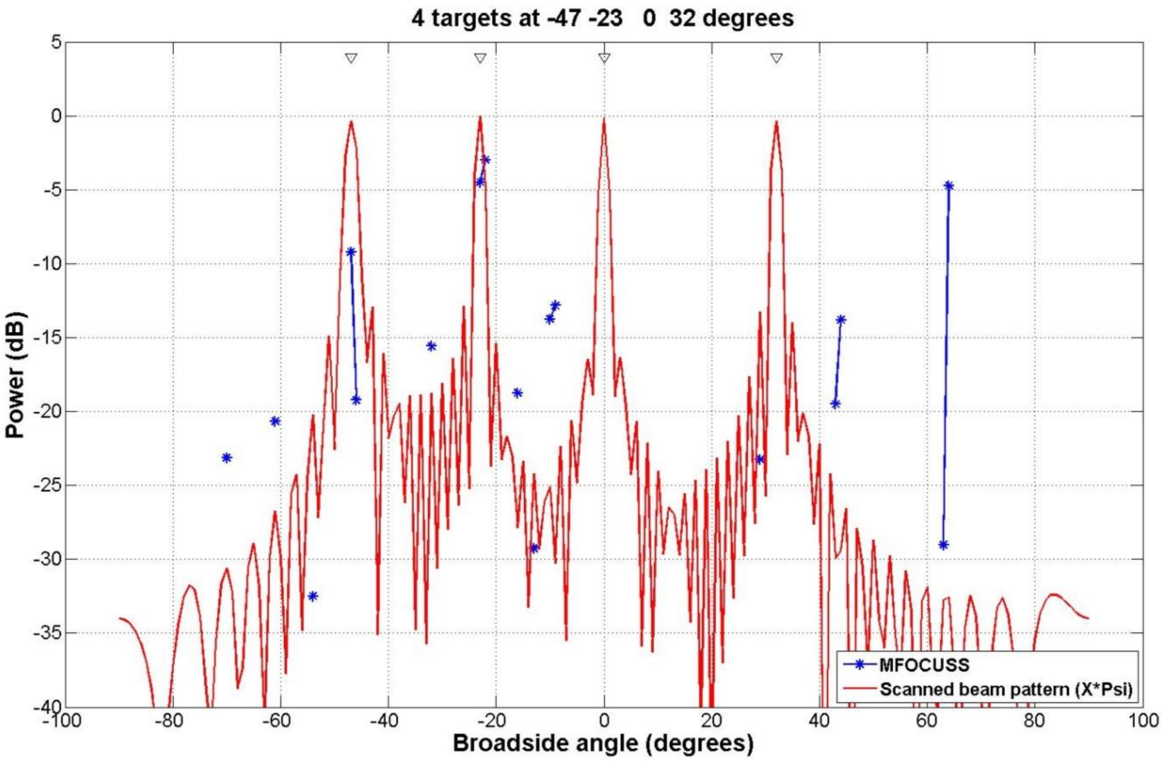

Figure 20: Example of a failed reconstruction for 4 targets without noise. The MSE for this reconstruction is 0.8445.

Unequal strength targets

In practice *K* targets are expected not to have the same amplitude. To see the effect of targets with varying strengths the same setup has been used but with random target strengths varying between 0 and -10dB. The results are summarized in Table 4, from which it can be seen that more solutions with incorrect columns are identified than in the case of targets with equal strength. Now already for 3 targets 0.5 percent of the reconstructions have one or more identified columns, which is below the theoretical limit of 3.85 targets. Also the average MSE increases for the situation with 3 targets since there are more incorrect identified columns found.

Table 4: Number of solutions with incorrect identified columns and average MSE for random DOA estimations of up to 6 targets. The targets have a random strength between 0 and -10dB

No of targets	1	2	3	4	5	6
Wrong solutions within 1000 experiments	0	0	5	51	231	505
average MSE	3.97E-10	1.66E-10	1.53E-03	0.0167	0.0816	0.195

An example of wrong reconstruction of 3 targets is shown in Figure 21. It shows that the 2 strongest targets are reconstructed at the correct DOA but the weakest target at approximately -6dB is not reconstructed. Besides that also a lot of erroneous solutions are found with a maximal strength of -14dB.

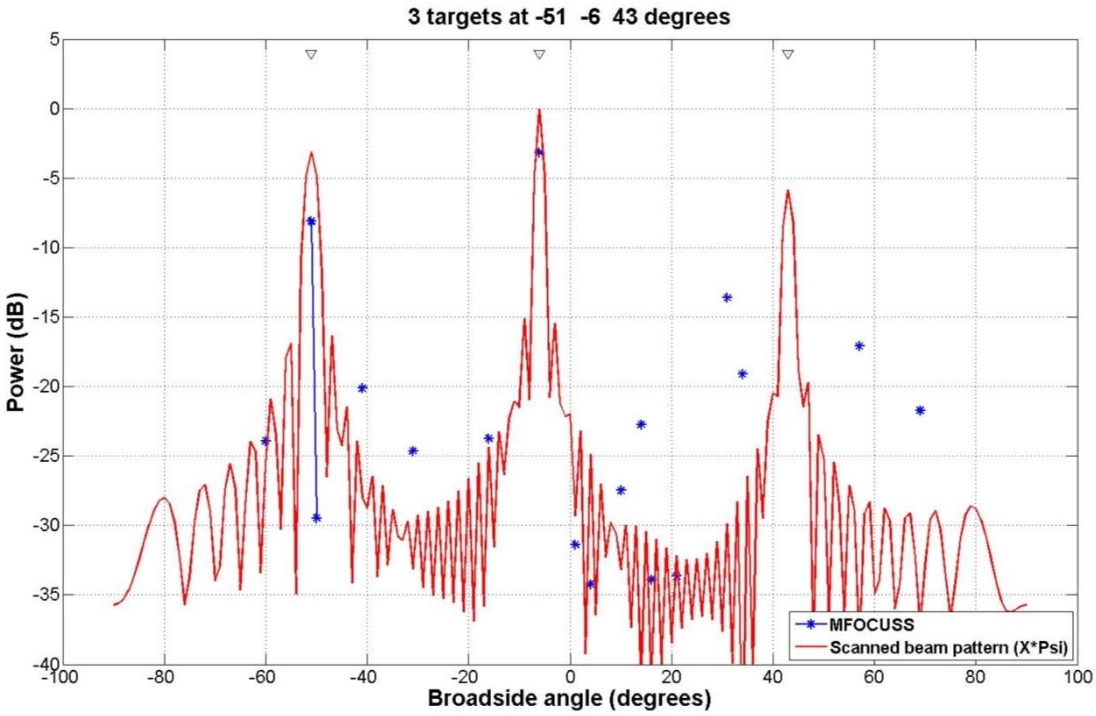

Figure 21: incorrect reconstruction of K=3 random strength targets with random DOAs. The targets have strengths -3dB, 0dB and -5.2dB (FLTR). λ=10^{-10} (equal to the noise variance). This incorrect solution is found because of the missing solution of the lowest strength target. The MSE for this reconstruction is 0.2893.

When there are targets with an unequal strength the performance slightly decrease. When there are one or two simultaneous targets, they can be reconstructed correctly but for 3 targets there is a significant increase in errors resulting in 0.5 percent failure. For *K*=4 targets the number of wrong reconstructions again increases compared to the equal strength target case: from 2.8 percent to 5.1 percent.

4.1.3 Single target off the grid

As stated in section 3.3 the discrete nature of CS assumes that targets lie exactly on grid points of the angular discretization grid. In real life this is obviously almost never true. When targets are not on the partition grid but in between two angular gridlines, the error of the solution increases, resulting in incorrect estimated DOAs [15]. Using a very fine grid can partly solve this problem but using a finer grid will also result in a higher computational effort and also the coherence in the CS problem can be affected. Hence it is important to see what the effects are when there are targets with DOAs which are not on the partition grid. Therefore various simulations, without noise, but with off grid DOAs have been carried out. The most relevant results are summarized in this section for a single target, and in section 4.1.4 for *K* targets.

It is expected that the closer the actual DOA is to the grid the less discretization noise arises. To verify this simulations have been done with different off grid distances. The results are shown in Figure 22 and Table 6.

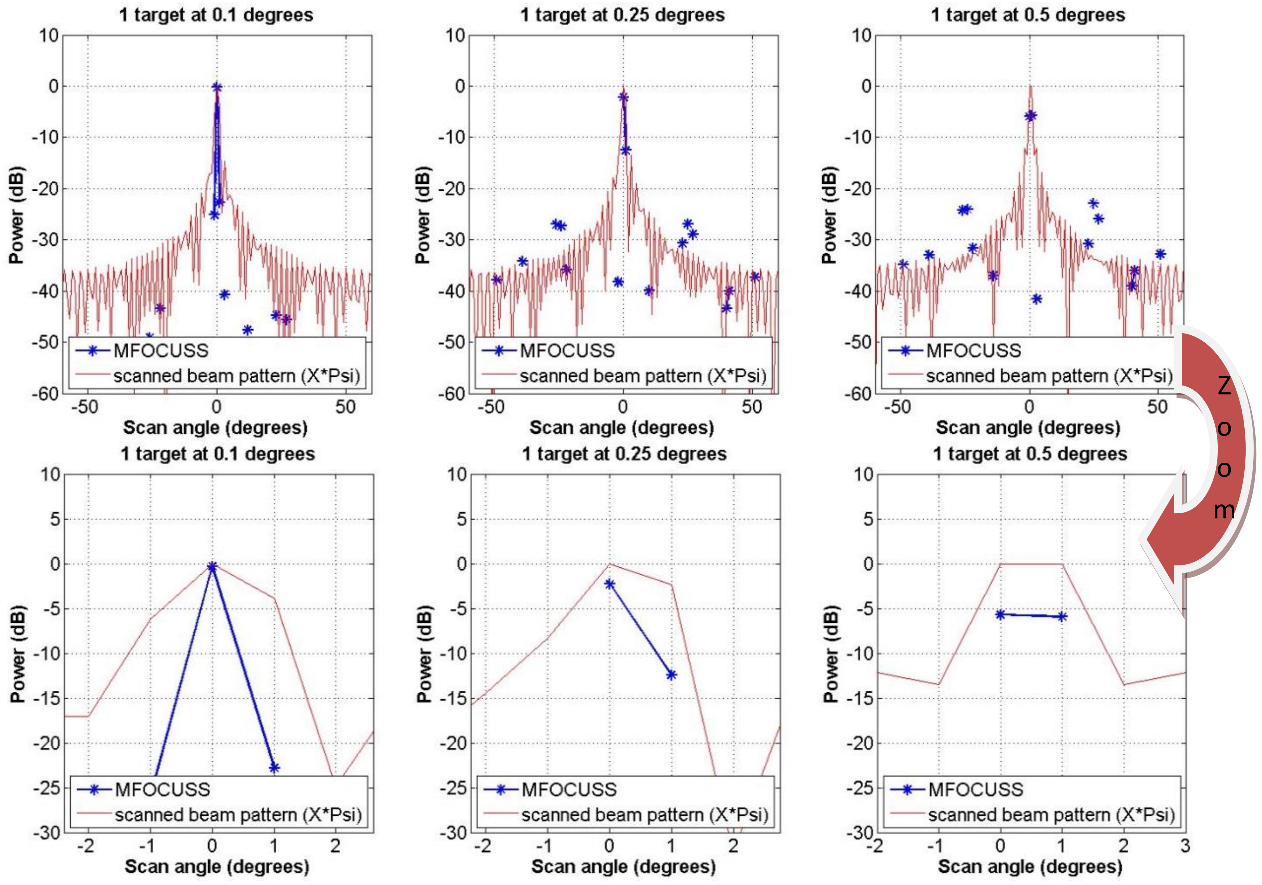

Figure 22: Targets with different off grid distances around boresight. No noise is added to the signal.

As expected the discretization noise increases as the targets are more off grid. In Table 5 the MSE and number of needed iterations are shown for several off grid distances. As expected it shows that the MSE increases for increasing off grid distances. The number of iterations needed not necessarily increase or decrease for increasing off grid distances, but they do increase significantly compared to the on grid cases.

Table 5: MSE and number of iterations needed for different off grid distances at boresight.

real DOA (degrees)	0.1	0.2	0.25	0.3	0.4	0.5
MSE	3.56E-04	5.19E-03	2.03E-02	2.62E-02	3.63E-02	4.02E-02
Iterations	105	89	76	91	160	139

Decrease in Power of the found solutions

Compared to the on grid situations in Figure 18 it is observed that the power level of the targets is decreased. This is due to the fact that the energy of the sparse solution is no longer a single solution but gets spread out over the two adjacent gridpoints. This can be seen in the zoomed in version of the three examples of Figure 22 and Figure 23: there are two solutions found at the adjacent gridpoints with a power of ≈-6dB, instead of 1 exactly at the grid point at 0dB. The power levels being ≈-6dB is however 3dB more than expected since it is expected that the off grid effect would result in two solutions with a power level of -3dB at adjacent gridpoints. (There is discretization noise which should be added to the total energy, but as can be seen in Figure 23 this is never higher than -22dB). The loss of ≈3dB power in the solution is attributed to a wrong normalization, when targets are off grid, within the algorithm. This was verified with additional simulations in which the amplitude of the found solution of an on grid target was manually decreased with 3dB and 6dB. The MSE for the original case was <10^{-10} since the solution is exact. The MSE for the -3dB case is in the order of the MSE found for 0.5 degree off grid signals. The MSE for the -6dB signal is in the order of double the MSE found for 0.5 degree off grid signals. Thus it can be concluded that the power levels are lower than expected, but this is purely visual. This normalization error only occurs for off grid signals and was not traced within the code during this thesis.

It has been investigated if the DOA has an effect on the discretization error. Therefore targets with different DOAs, with the maximal off grid distance of 0.5 degrees, have been simulated. In Figure 23 and Table 6 three examples are shown.

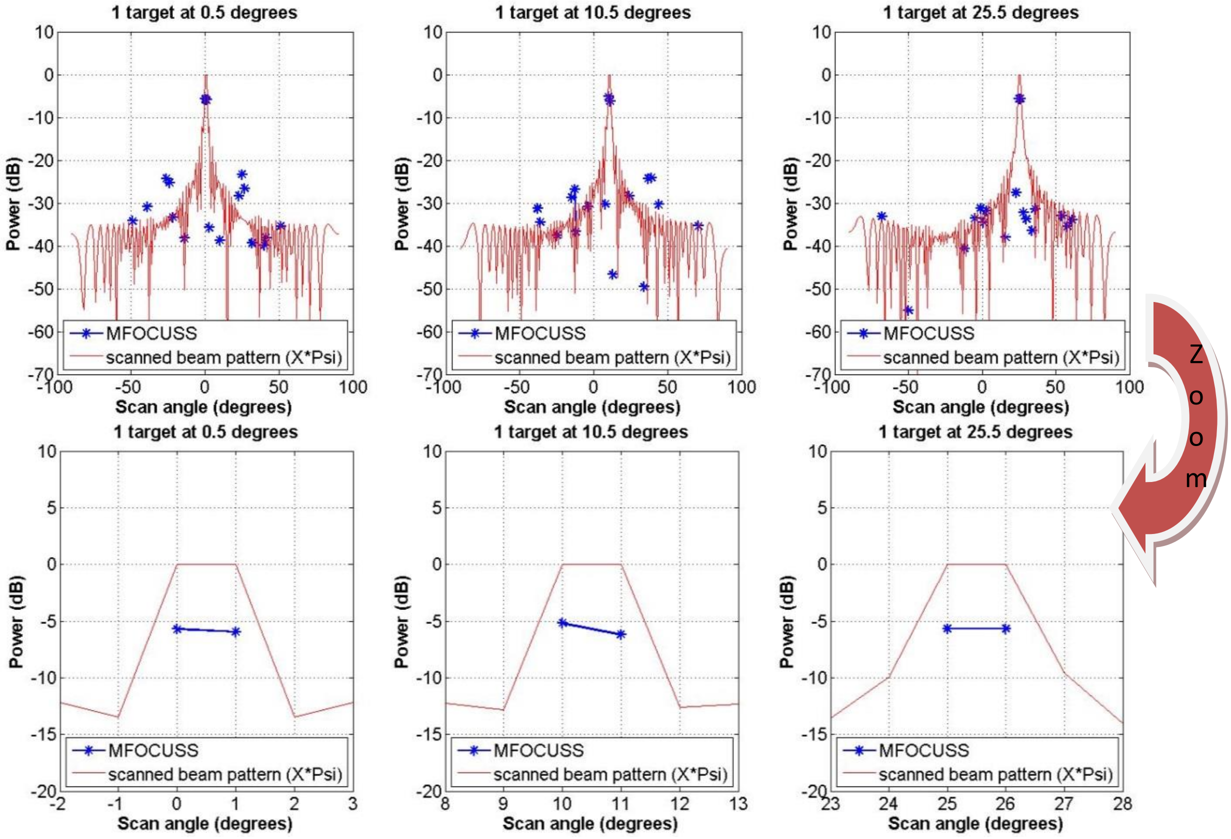

Figure 23: Targets with different DOA's which are in the middle of two gridlines. No noise is added to the signal.

The MSE indicates that the discretization noise is larger for DOAs closer to boresight as can be seen in Table 6. The MSE decreases when the DOA is further off boresight. In Figure 24 it can be seen what the result is of the discretization noise: wrong solutions at lower power levels around -25 and 25 degrees for the target at 0.5 degrees. The discretization noise in the reconstructed signal, results in higher sidelobes in the angular spectrum around -25 and 25 degrees. The spectrum of the original signal has no high sidelobes around these angles. The reconstruction of the target at 25.5 degrees shown no sidelobes and matched the original signal better, as can be seen in Figure 24.These sidelobes increase the MSE and give rise to the discretization noise as shown in Figure 23.

Table 6: MSE and number of iterations needed for different off grid DOAs.

real DOA (degrees)	0.5	10.5	25.5
MSE	0.037	0.028	0.010
Iterations	89	82	242

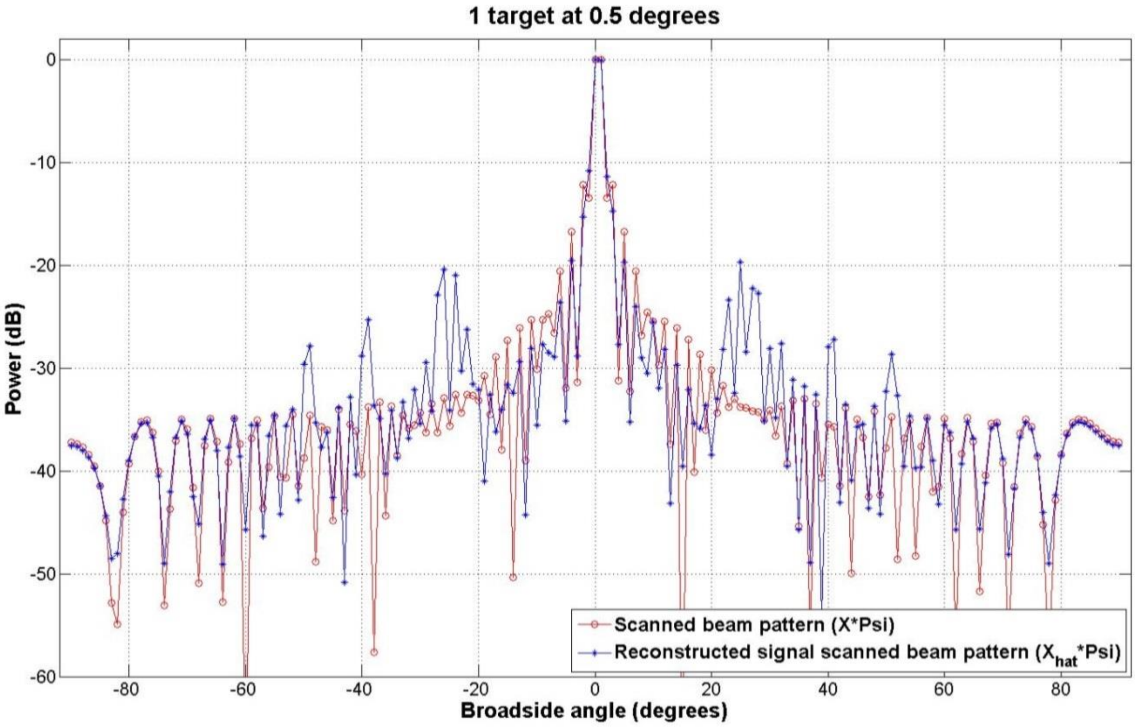

Figure 24: Reconstructed beam pattern and original beam pattern for off grid signals with DOA 0.5. The MSE is 0.037 for this case.

4.1.4 *K* targets off grid

The effect of discretization errors on the relation between the needed number of working elements and the number of targets to be resolved is studied below. In these simulations the simultaneous presence of *K* targets has been simulated for random off grid DOAs over 1000 different runs. Since the effect of the discretization errors increased with the off grid distance, a maximal off grid shift of 0.5 degrees has been used on the angular grid with one degree resolution. When the targets are 0.5 degrees off grid this also means that they are in the middle of two grid points. Hence in this case a correctly identified column is one of the two adjacent columns in which the target is supposed to arrive. Normally, in the case of off-grid targets and grid-based reconstructions, you should have a clustering process after which you evaluate the correctness of the solution. Two adjacent solutions would then be clustered together in one solution.

The results are shown in Table 7. The discretization errors cause 2 additional wrong reconstructions for *K*=3 incoming targets, compared to the on grid scenario of Table 3. Besides the additional errors for 3 targets the MSE is also higher for every number of present targets due to the discretization errors, but they are in the same order of the result found in Table 6. This indicates that besides the increase in MSE due to the off grid effects, there is not much additional error. For *K*≥4 targets the MSE significantly increases compared to the result from Table 6, indicating that the number of wrong reconstructions are increasing.

Table 7: Number of solutions with incorrect identified columns and average MSE for random DOA estimations of up to 6 equal strength targets which are0.5 degree off grid.

No of targets (*K*)	1	2	3	4	5	6
Wrong solutions within 1000 experiments	0	0	2	22	176	426
Average MSE	1.36E-02	2.11E-02	5.20E-02	1.33E-01	3.07E-01	4.89E-01

Examples of a correct and wrong solutions are shown in Figure 25 for *K*=3 targets. In the example with the correct solution clearly there are solutions found on the correct adjacent gridpoints. The other low level solutions found are due to the discretization errors, and they are at least 12dB lower than the solutions for the targets. This means the targets can easily be identified. In the example with the failed reconstruction, the target with DOA 46.5 degrees there are no solutions found in the corresponding (adjacent) columns of the $\mathbf{P}_y$ vector. Thus this solution is marked as wrong. It should also be noted that for the solutions found at the correct DOA are at a lower power level which can be up to a level of -6dB. This is due to the discretization errors which spread the energy over the adjacent gridpoints and wrong noisy solutions and the normalization error discussed in section 4.1.3.

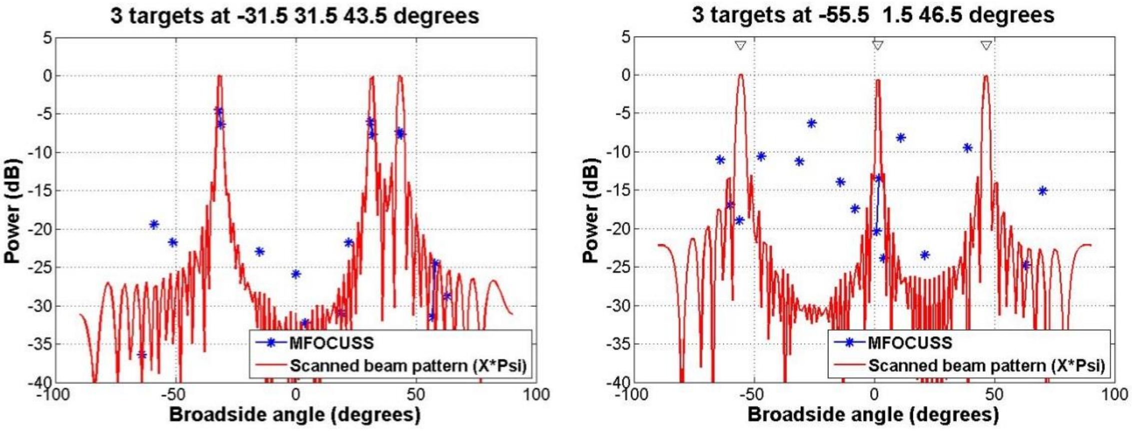

Figure 25: Example of a correct (left) and failed (right) reconstruction. The examples are taken from out of the 1000 reconstructions when 3 equal strength (0dB) targets with random DOAs arrive.

Table 8 shows simulation results for targets with a random strength between 0 and -10dB. For *K*=3 incoming signals the number of wrong solutions are increased up to nine, while MSE levels are around the same level as for the equal strength cases in Table 7. This indicates that the unequal strength signals do cause some additional wrong reconstructions, but on average the MSE is around the same level.

Table 8: Number of solutions with incorrect identified columns and average MSE for random DOA estimations of up to 6 targets which are 0.5 degree off grid. The targets have a random strength between 0 and -10dB

No of targets	1	2	3	4	5	6
Wrong solutions within 1000 experiments	0	0	9	77	293	587
Average MSE	1.41E-02	2.09E-02	5.17E-02	1.21E-01	2.58E-01	3.93E-01

An example of a successful reconstruction for maximal off grid targets with random strengths is shown in Figure 26. The solutions found are all ≈8dB above the wrong solutions found due to the discretization errors.

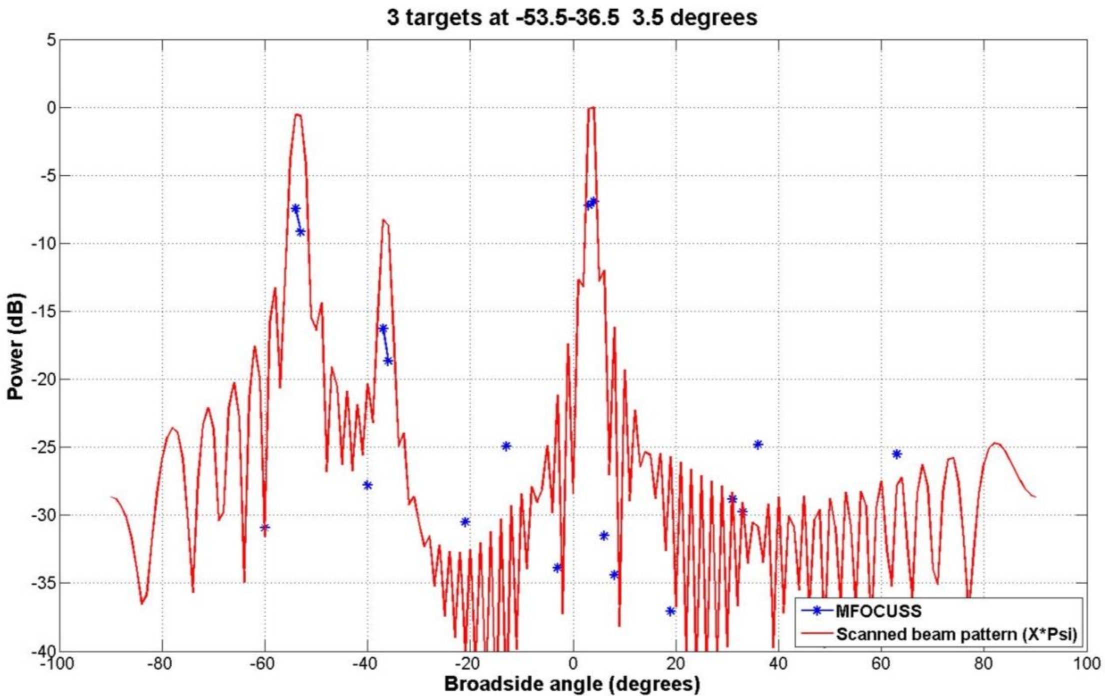

Figure 26: Example of a correct reconstruction when maximal off grid targets having random varying strengths between 0 and -10dB. In this case the targets strengths are FLTR: -0.5dB, -7.9dB and 0dB and $\lambda=10^{-10}$.

4.1.5 Analysis and discussion: recovery without noise

For on grid single target cases, CS with the MFOCUSS algorithm is perfectly capable to estimate the DOA of a target correctly. When the number of targets K is increased the probability of correct DOA estimation decreases since the received signal becomes less sparse. The theoretical limit of $K \log(N)$ working elements needed to identify K incoming targets was verified with simulations for the noiseless case with equal strength targets. For unequal strength targets the number of wrong reconstructions for K=4 targets doubled, and also for $K=3$ an additional 0.5% of the reconstructions failed. Hence it can be said that targets with different strengths have a degrading effect on the performance of CS.

When targets are not exactly on the discretization grid, discretization noise increases as the off grid distance of the target increases. The MSE of the found solutions increases significantly to $\approx 4*10^{-2}$ due to discretization errors. Also the number of iterations needed by the algorithm needed to find a solution increases significantly. Discretization errors can hence not simply be neglected and they have a large influence on the performance.

The plotted results for maximal off grid distances (0.5 degree) show power levels that are lower than expected. This is purely visible and it is highly likely that this is due to a normalization error within the algorithm. It only occurs for off grid signals and was not traced within the code during this thesis.

4.2 Recovery within noise

In real life situations the received signal is deteriorated by noise. This noise can have degrading effects on the recovery of targets using CS depending on how robust the algorithm is against noise. To determine the robustness against noise, simulations with white noise added to the received signal have been done. In these noisy simulations the regularization parameter λ is set at the value of the noise variance.

4.2.1 Single target on the grid

When there is no noise the found solutions are exact as can be seen in Figure 18. When noise is present the reconstruction is also noisy. In Figure 27 the results for the recovery of the same random targets as in Figure 18 are shown, but now with an input SNR of 20dB. Where in the absence of noise the solutions where exact now the solutions are disturbed because of the noise. The noise in the solution never exceeds the noise level. The targets can still easily be identified since the amplitude of the solutions found at DOAs of the target is much higher than the amplitude of the noisy solutions.

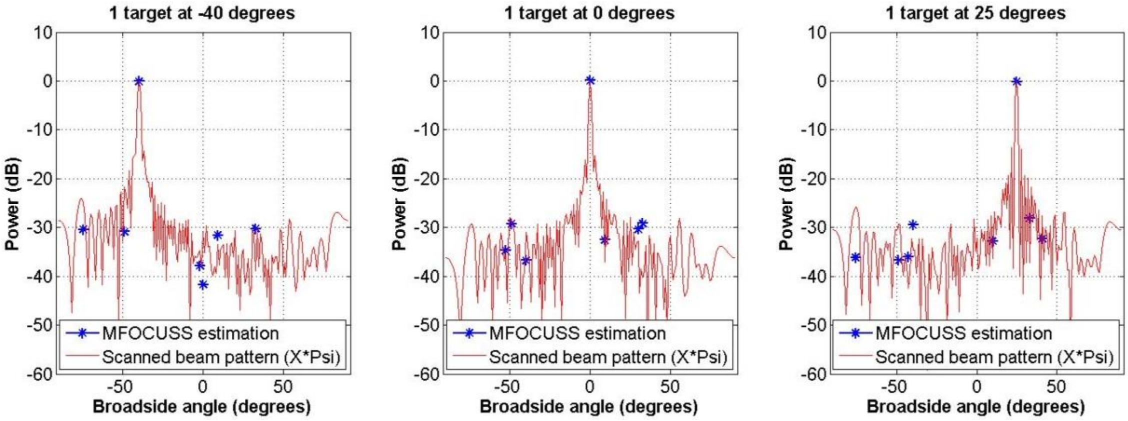

Figure 27: recovery of a single target on the grid in the presence of noise. The noise level is at -20dB.

In Table 9 the MSE and the number of iterations needed are shown. Due to the noise the MSE is increased. The increase of the MSE can also be observed in Figure 28. Although there are some clear deviations from the original signal, the reconstructed signal does not have exceptional large deviations or sidelobes which could result in high power wrong solutions. The number of iterations increases since the noise makes an exact reconstruction not possible, hence the algorithm has to use more iterations to reach the termination criteria.

Table 9: MSE and number of iterations needed for reconstructing (single) targets with a -20dB noise level.

DOA (degrees)	-40	0	25
MSE	1.17E-02	1.24E-02	1.62E-02
Iterations	174	84	72

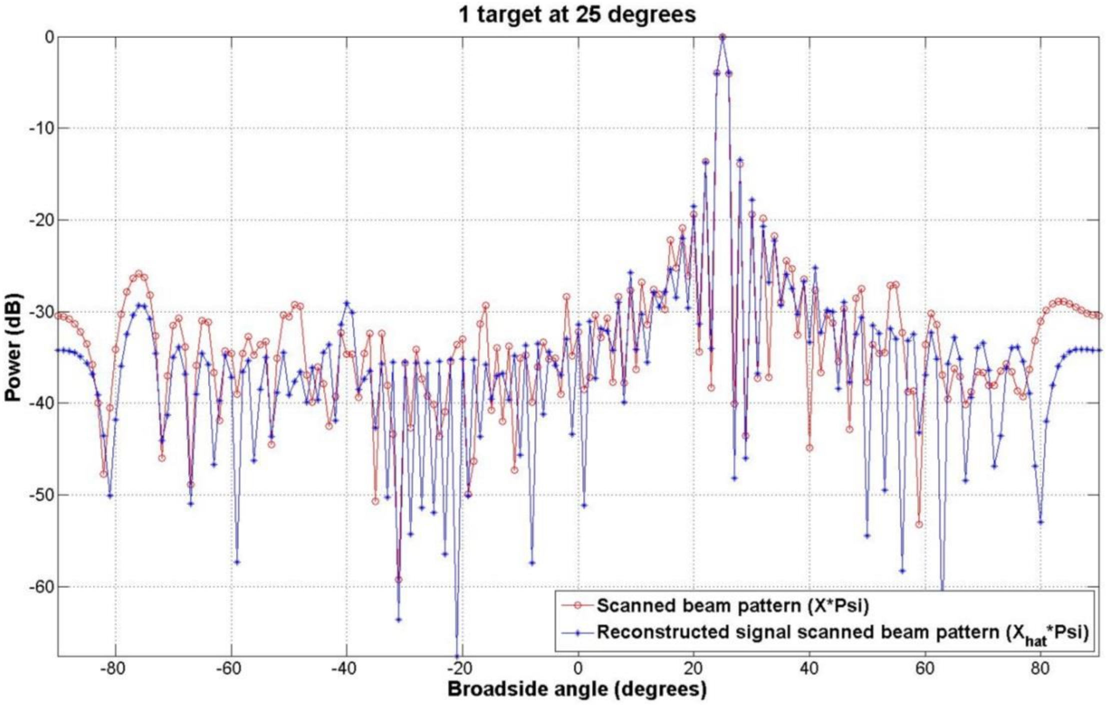

Figure 28: Scanned beam pattern and reconstructed scanned beam pattern for the reconstruction of a single target with a DOA at 25 degrees within -20 dB noise

4.2.2 *K* targets on the grid

In the first place, to be able to relate the results for the MSE to a correct value, the MSE for different noise levels for a single on grid target have been calculated. The results can be found in Table 10 and give an indication of MSE levels for correct reconstructions in the presence of noise.

Table 10: MSE for different noise levels for a single on grid target.

Noise level [dB]	-100	-50	-40	-30	-20	-10	0
MSE (Doa =0 deg)	1.42e-10	1.42e-05	1.30e-04	1.30e-03	1.23e-02	0.1132	0.51

Reconstructing of *K* unequal strength targets is closer to reality than reconstructing *K* equal strength targets. Results for equal strength targets are expected to be better than those of unequal strength signals. Hence only extensive simulations have been done for various noise levels with *K* unequal strength targets at random DOAs between -60 and 60 degrees. Again double DOAs in a single run are not possible. The target strengths again vary between 0 and -10dB and the noise levels are between -30 and -10dB. The results for 1000 runs per noise level with *K* targets are shown in Table 11. “Wrong reconstructions” are again the reconstructions in which not all columns corresponding to the real DOA are correctly identified. From Table 11 it can be seen that the MSE increases with the increasing noise according to the values found in Table 10. The MSE for *K* targets is always in the order of the noise level. However it should also be noted that the average MSE is often higher for a lower number of *K* with only correct solutions. This makes it hard to give a boundary value for the MSE that indicates only correct or incorrect solutions. The average MSE can give an indication if the results are in line with the MSE expected for the noise level, but it is better to look at the number of incorrect identified columns to get an indication of the success of the reconstructions. Of course the MSE says something about the error in the reconstructed signal $\hat{\mathbf{X}}$, but if this is error is higher it does not directly mean that there is also an error in DOA estimation.

Table 11: Results for Unequal strength on grid targets for various noise levels. The targets vary in strength between 0 and -10dB. For every noise level and every number of *K* targets 1000 runs have been simulated.

Number of targets *K*		1	2	3	4	5
-30 dB Noise level	wrong reconstructions	0	0	6	53	244
	Avg. MSE	4.39E-03	2.17E-03	2.52E-03	1.69E-02	8.34E-02
-20 dB Noise level	wrong reconstructions	0	1	10	69	268
	Avg. MSE	4.24E-02	2.28E-02	1.35E-02	2.95E-02	9.51E-02
-10 dB noise level	wrong reconstructions	7	47	75	197	464
	Avg. MSE	2.93E-01	2.00E-01	1.32E-01	1.42E-01	2.16E-01

An example of a correct reconstruction with *K*=4 random strength targets within -20dB noise is shown in Figure 29. Since the signals are on grid the reconstruction is almost exact. Only the noise causes some small deviations in amplitude and in angle for the target with a DOA of 18 degrees.

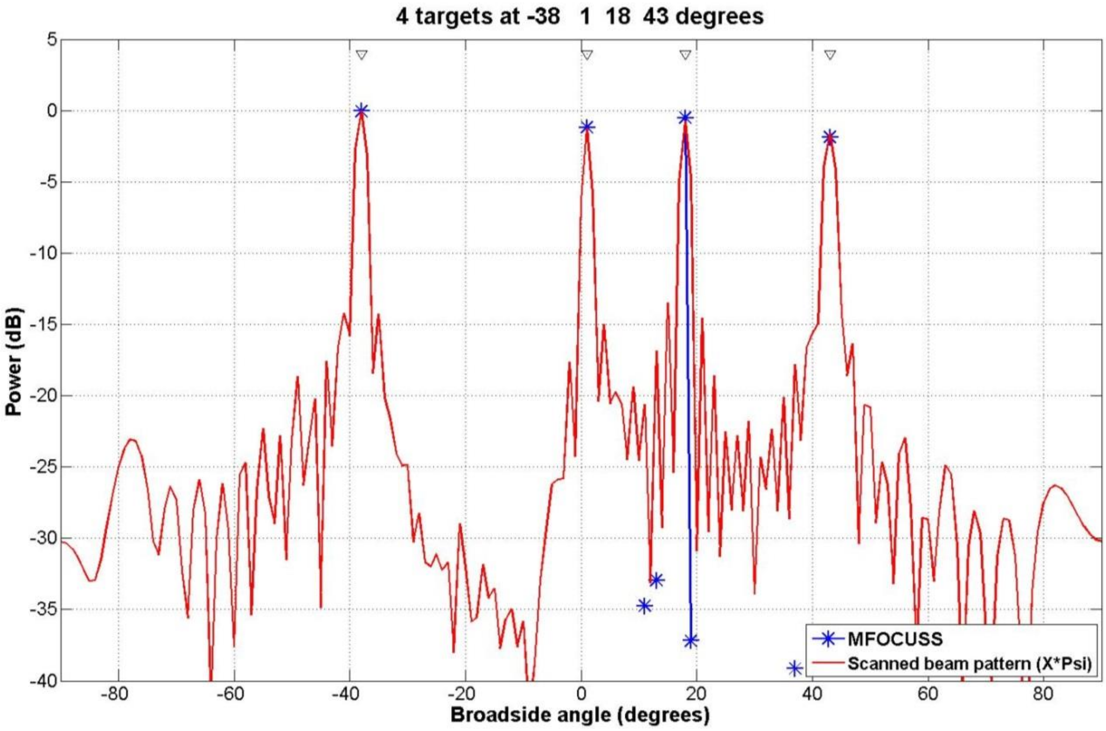

Figure 29: Example of a correct reconstruction for 4 random strength on grid targets with a -20dB noise level. The signal levels in this case are, from left to right: 0dB, -0,96dB, -0.45dB and-1.55dB. The MSE for this reconstruction is 0.0049.

For the noise level of -20dB there is a single wrong reconstruction for *K*=2 targets in the simulations that have been performed. This wrong reconstruction is shown in Figure 30. It can be seen that the reconstruction finds two solutions at -15dB, which is ≈5dB below the original signal strength. The solutions are found at a DOA of -56 and -58 degrees. Because the true DOA is at 57 degrees, it is counted as a wrong reconstruction. As stated before, this ambiguity would be solved by a clustering function.

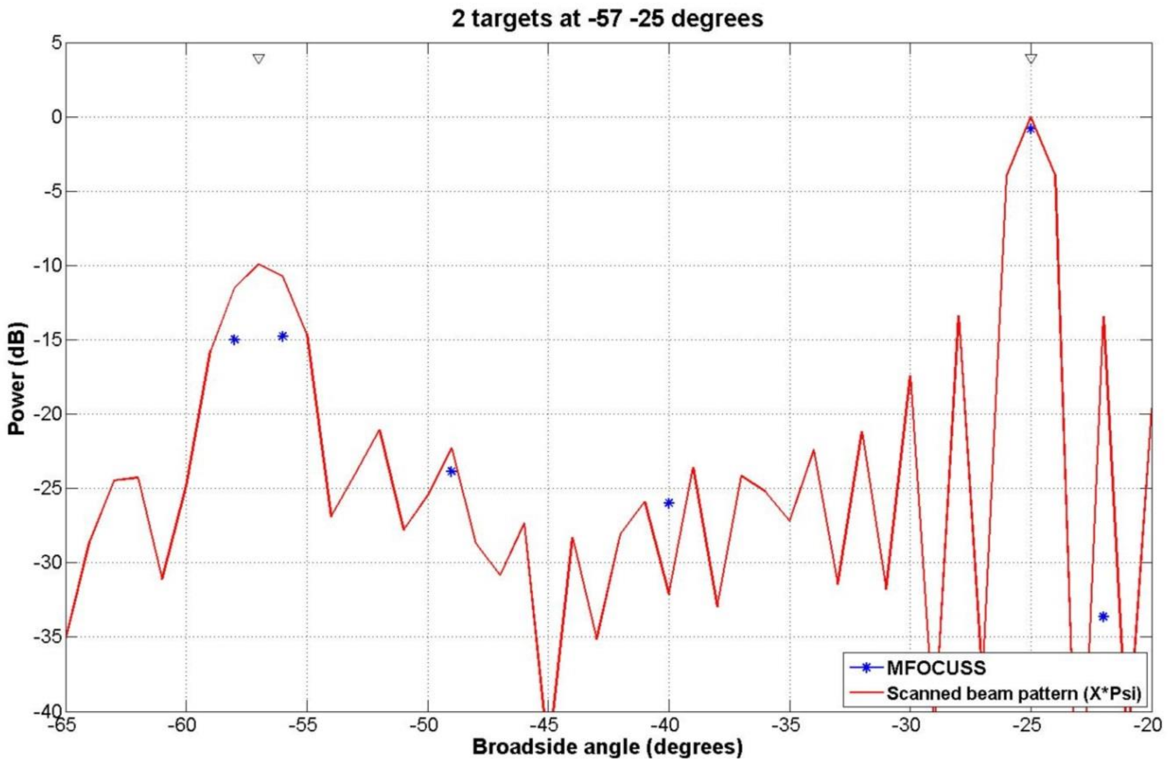

Figure 30: The wrong reconstruction for 2 on grid targets with a -20dB noise level. The solutions at -56 and -58 degrees at -15dB are found instead of a solution at 57 degrees at 0dB. The original signal levels are at -10 and -0.45dB. The MSE for this case is 0.0163.

When the noise level is at -10dB also reconstructions for a single target fail. This makes sense since the random target strengths are between 0 and -10dB and thus the target can be at the same level as the noise. This makes the overall received signal not sparse anymore, causing the reconstruction to fail. This emphasizes the need for a SNR>10dB when we want to use CS for DOA estimation.

4.2.3 *K* targets off the grid

Not only noise has a degrading effect, also the discretization errors. To see what the combined effect is, simulations have been done with *K* off grid targets within the presence of noise. Again the targets are at the maximal off grid distance of 0.5 degrees. A target is thus in between two columns, making a correctly identified column one of the two of these columns. To get an idea of the MSE for off grid signals in different noise levels, the MSEs are calculated for single simulations with 0.5 degree off grid signals. They are shown, for convenience with the on grid MSEs, in Table 12. In Table 12 it can be seen that the off grid effects dominate the MSE up to noise levels of -10dB.

Table 12: MSE for different noise levels for on grid and 0.5 degree off grid signals.

Noise level [dB]	-100	-50	-40	-30	-20	-10	0
On grid MSE	1.42e-10	1.42e-05	1.30e-04	1.30e-03	1.23e-02	0.1132	0.51
Off grid MSE	0.037	0.039	0.042	0.0374	0.044	0.149	0.547

The results for varying noise levels are shown in Table 13. There is a slight decrease in performance compared to the situation where only noise was present. Remarkably enough there are fewer wrong reconstructions for a -10dB noise level for the off grid situation. This might be due to the fact that for the off grid situation the two adjacent columns count as a correctly identified column. For the “noise only" situation only the correct column results in correct reconstruction. And as was seen in Figure 30 also noise can cause the solution to be spread in angle and thus onto adjacent gridlines.

The MSE is increased for the off grid case which means there is a larger error in the reconstruction when the target is also off grid. This is in line with the results from Table 12, but again the MSE can be lower for an increasing number of *K* targets and noise levels ≥20dB. This again states that, for the DOA estimation results, the MSE is a good indication of the quality of the solutions found, but that there is no hard boundary value for the MSE which indicates wrong reconstructions.

Table 13: Results for Unequal strength off grid targets for various noise levels. The targets vary in strength between 0 and -10dB and arrive 0.5 degrees off grid. For every noise level and every number of *K* targets 1000 runs have been simulated.

Number of targets *K*		**1**	**2**	**3**	**4**	**5**
-30 dB Noise level	wrong reconstructions	0	0	9	76	306
	MSE	1.75E-02	2.19E-02	4.71E-02	1.18E-01	2.58E-01
-20 dB Noise level	wrong reconstructions	0	1	13	93	330
	MSE	5.99E-02	4.90E-02	5.95E-02	1.34E-01	2.69E-01
-10 dB noise level	wrong reconstructions	3	29	66	212	470
	MSE	3.23E-01	2.58E-01	2.05E-01	2.74E-01	3.67E-01

4.2.4 Analysis and discussion: recovery within noise

The noise level in the found solution never exceeds the original noise level and hence it can be concluded that noise does have effect on the found solution but at least it is not increased by the algorithm. When the SNR is ≥20dB, the discretization errors seem to dominate the errors in the reconstruction.

Accounting for both noise and discretization errors it have been observed that not *M*/log(*N*) = **3.85** targets can be recovered. For an SNR of 30dB the number of targets that can be reconstructed is more in the order of 2 to 3 targets.

In some exceptional cases it was observed that for on grid (weaker) targets, solutions where found on both the adjacent gridpoints. The "correctly identified columns" being the criteria for a correct reconstruction, these exceptional cases are marked as a wrong reconstruction. A clustering algorithm for the hits that are found at contiguous grid cells should solve this. This again shows the importance of a good detection algorithm and an SNR≥10dB.

The average MSE can be used to get an indication of the found solutions and especially the error in the reconstructed signal $\widehat{\mathbf{X}}$. An error in $\widehat{\mathbf{X}}$ however does not always mean that the DOA estimation is incorrect. This makes it hard to use a value for the average MSE as a boundary. It is better to look at the combination of the incorrect identified columns and the average MSE to judge the quality of reconstruction.

4.3 Using Multiple Measurement Vectors

The example of a wrong reconstruction in Figure 31 shows a -11dB solution at ≈-20 degrees purely due to the discretization errors. These errors can in real life result in a false detection. The MFOCUSS algorithm is capable of working with MMV which could improve results. Assumed is that the time snapshots have a common sparsity profile so that the indices of the nonzero entries are independent of the time snapshots used, see also section 3.2.5. Simulations have been done to obtain insight in these improvements with the use of multiple time snapshots. The simulation results can be used to see if they can fit the demands of a real life system.

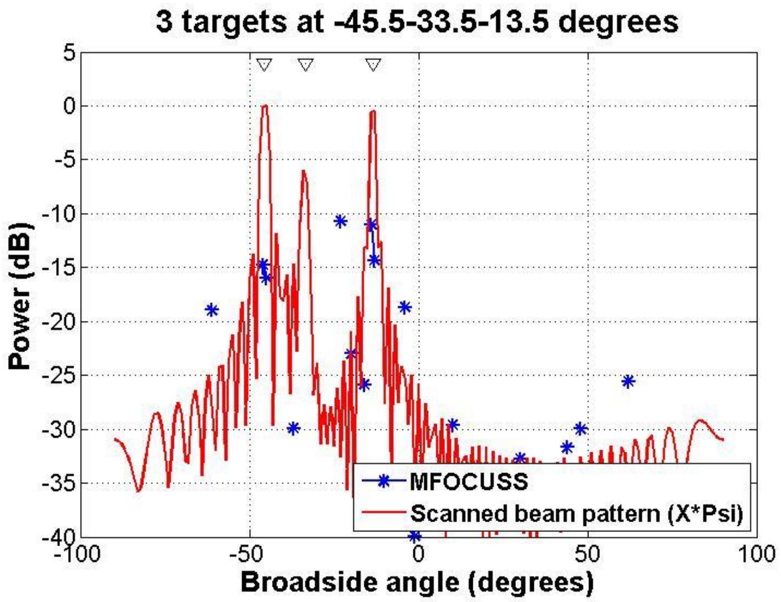

Figure 31: Example of a wrong reconstruction of 3 random unequal off grid signals without noise.

4.3.1 Results within noise with off grid targets

The example shown in Figure 31 has been reconstructed using three and ten time samples. Additionally the noise level has been set to -20dB, to get closer to the real life situation. The result is shown in Figure 32. When three time samples are used instead of one, the wrong solution caused by the discretization errors diminishes. The overall power level of noise in the reconstruction decreases to -25dB or lower. Additionally there are no other high level wrong solutions and the target at -33.5 degrees is also reconstructed. This indicates that the use of three to ten time samples can already improve the results significantly.

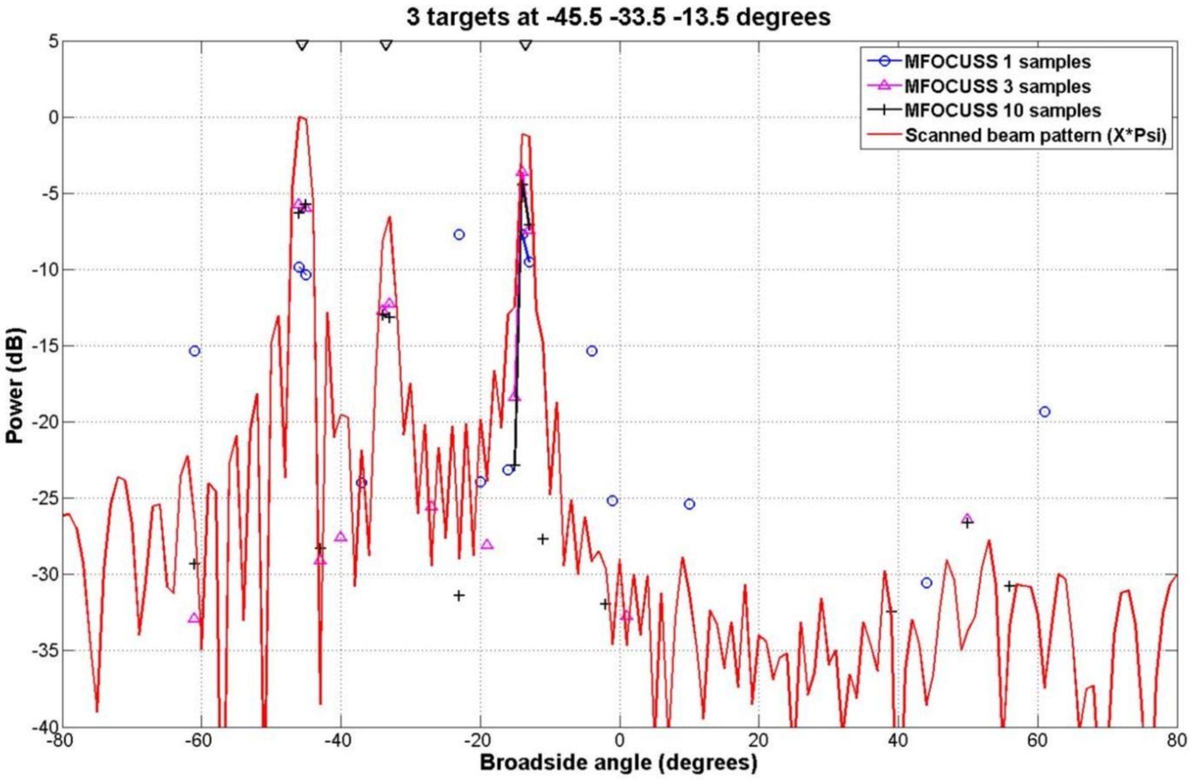

Figure 32: Reconstruction of 3 off grid signals within -20dB noise with 1,3 and 10 time samples used.

To see the effects with more statistical significance, 1000 runs are performed with *K* targets having random off grid DOAs for different noise levels. 1,3 and 10 time samples have been used to reconstruct the signal. (Obviously the results using 1 time sample are the results obtained in the previous simulations presented in section 4.2.3). The results for the solutions with incorrect identified columns are shown in Figure 33. Note that the number of incorrect solutions is presented on a logarithmic scale to better emphasize the point where the first reconstructions start to fail. The improvements of using MMV can be observed quite easily: when 3 time samples are used for noise levels of -30 and -20dB all 1000 reconstructions are correct up to 4 targets are correct. For a noise level of -10dB there is 1 error for *K*=3 targets, and 5 errors for *K*=4 targets. When 10 time samples are used, for all noise levels all reconstructions are correct as long the number of targets *K*≤4. When *K*>4 the number of incorrect reconstructions start to increase for a noise level of -10dB and when *K*>5 for a noise level of -20dB.

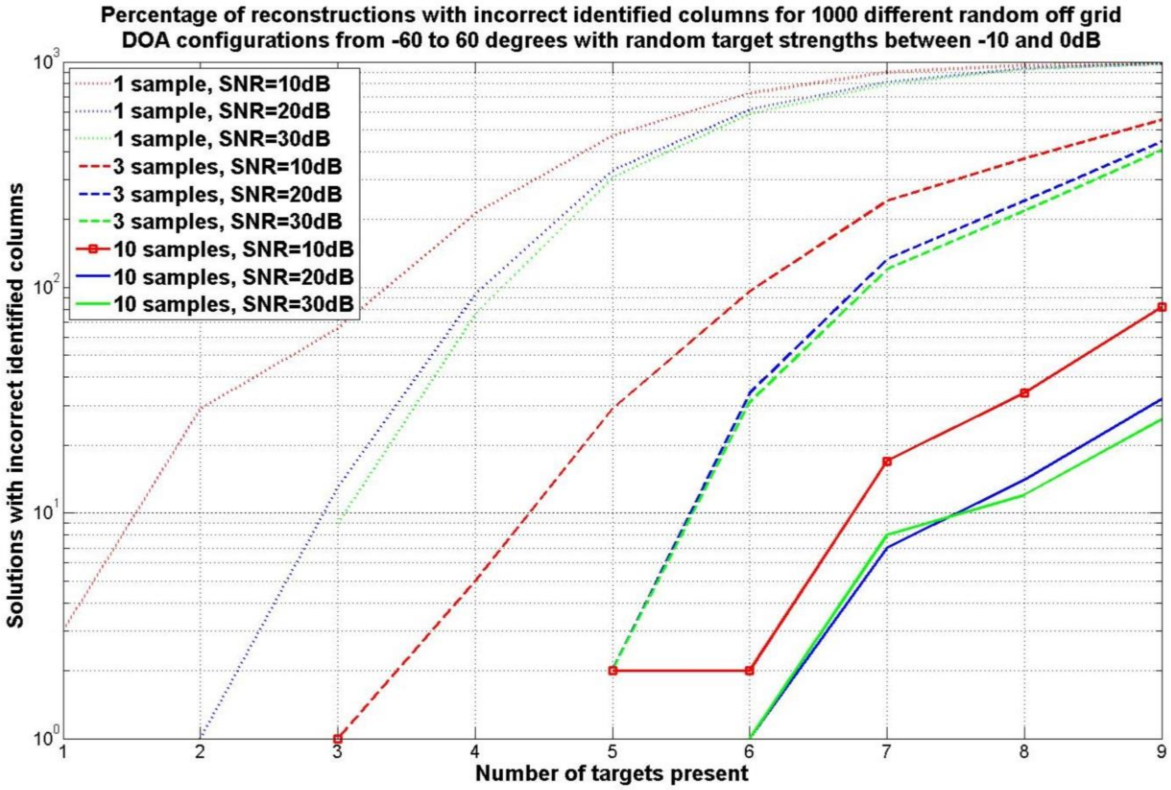

Figure 33: Results for *K* targets at random off grid DOAs, for different number of time samples used. The targets have a random strength between -10dB and 0dB.

There is however 1 exception for the -10dB noise level and 3 targets whereas also 1 erroneous reconstruction seems to appear. This reconstruction is shown in Figure 34 where it can be seen that the 3 targets are very close to each other. The middle and weakest target is therefore missed in the reconstruction, since there are no solutions found at -46 and -45 degrees. This is however not remarkable when one looks at the original scanned beam pattern. Since the middle target is the weakest it is more likely that two targets are interpreted than three.

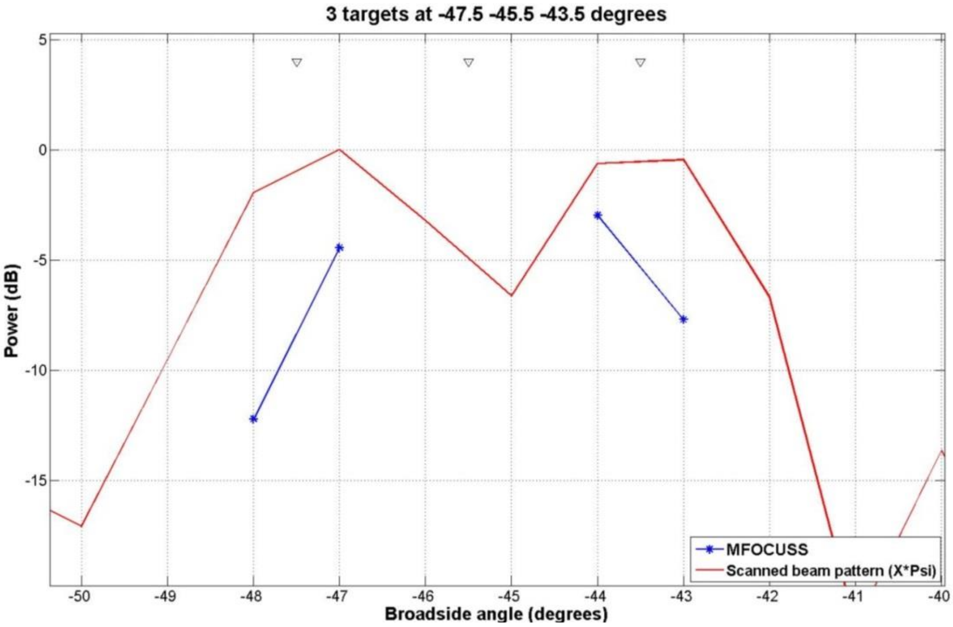

Figure 34: The erroneous reconstruction for 3 random targets within -10dB noise when 10 time samples are used.

4.3.2 Analysis and discussion

When multiple time samples are used the time samples must have the same sparsity profile. A rule of thumb is not to use more time samples than the number of available working elements. It is verified by simulations that using less than *M* time samples increases the results significantly. When 10 time samples are used the condition K=M/(log(N)) is valid again for SNR≥10dB. When 3 time samples are used the condition is valid again for SNR≥20dB. These increased results come at the price of a larger computational effort since the dimensions of the measurement and true solution vector increase. (The [*M* x 1] measurement vector **y** becomes an [*M* x *T*] measurement matrix and the [*N* x 1] true solution vector **x** becomes a [*M* x *T*] true solution matrix).

It is still an open question (an beyond the scope of this thesis) if and where the ability to use MMV can be integrated in the Radar processing chain. It is however clear that a lot of unwanted and distorting effects of noise and discretization errors can be decreased significantly by using more time samples.

4.4 Accuracy and Resolution

The angular accuracy and angular resolution of a phased array antenna are important practical performance parameters. Simulations must show if practical results can approach the theoretical values. The results related to the accuracy and resolution that has been obtained by simulations are presented and compared to the theoretical values in this section. Individual results might achieve high angular accuracy and resolution within random noise, but this is not guaranteed for all situations. Therefore 1000 simulations with (different) random noise must show the deviations between the individual solutions and give results with more statistical significance.

4.4.1 Accuracy: one target

According to equation (13) the theoretical angular accuracy for a 64 element ULA can be determined for different SNR levels, resulting in the maximal obtainable angular accuracy values as shown in Table 14. For the CS DOA estimated signal of the 16 element spare array, the accuracy is evaluated by simulating a single target for different noise levels, angular grid spacings and values for *λ.* The most relevant results related to the angular accuracy are presented.

Table 14: Maximal theoretical angular accuracy for different SNR levels for the 64 element ULA (using equation (13)).

SNR [dB]	Angular accuracy [deg.]
10	0.22
20	0.07
30	0.02

As previous results have shown, noise causes spread in amplitude and angular accuracy of the solutions. Discretization errors cause a spread in angle and amplitude, since the energy is spread between the two adjacent gridpoints and they can give rise to high power sidelobe like solutions not close to actual targets. Taking this in mind one could choose a fine discretization grid. In that case the targets always will be closer to the grid points, resulting in a small angular discretization error. An example of this is shown in Figure 35 where the decrease in grid spacing from 1.0 to 0.25 degrees results in a correct reconstruction instead of a wrong one. Note that for the 0.25 degree grid the signals are arriving also exactly in the middle of two gridpoints, being 0.125 degrees off grid.

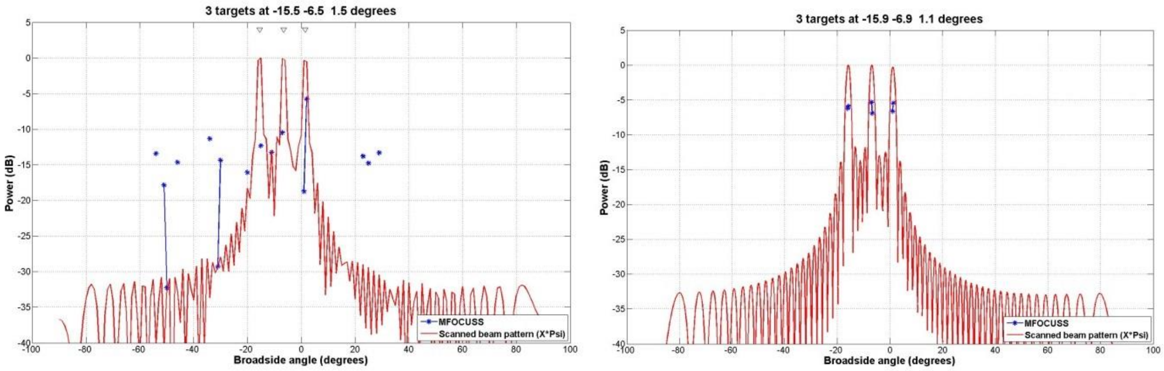

Figure 35: Example of a wrong reconstruction with maximal off grid (0.5 degrees) DOAs on a 1 degree spaced grid, due to discretization noise (left). Example of a correct reconstruction with maximal off grid (0.125 degrees) DOAs, on a 0.25 degree spaced grid (right). There is no noise in both reconstructions and λ=10^{-10}.

The grid spacing cannot be decreased unlimited and keep improving results continuously. When the grid size is chosen to be very small the number of columns of the sensing matrix will increase, making the sensing matrix highly rectangular. These large rectangular matrices are hard to invert and have a high coherence between the columns. This result in an increase in computational load and increases chance of numerical instability. For single on grid targets within -20dB noise the reconstruction was always exact on the 1 degree grid (see Table 11). Decreasing the grid spacing to 0.1 degrees, again 1000 simulations for a single on grid target with -20dB random noise and λ=0.01 have been executed. The results are shown in Figure 36. The smaller grid spacing results in a spreading for some of the solutions of the single (on grid!) target. This is due to the large coherence between the columns of the sensing matrix. Also note that the noise level lies around -22dB (Figure 36 left).

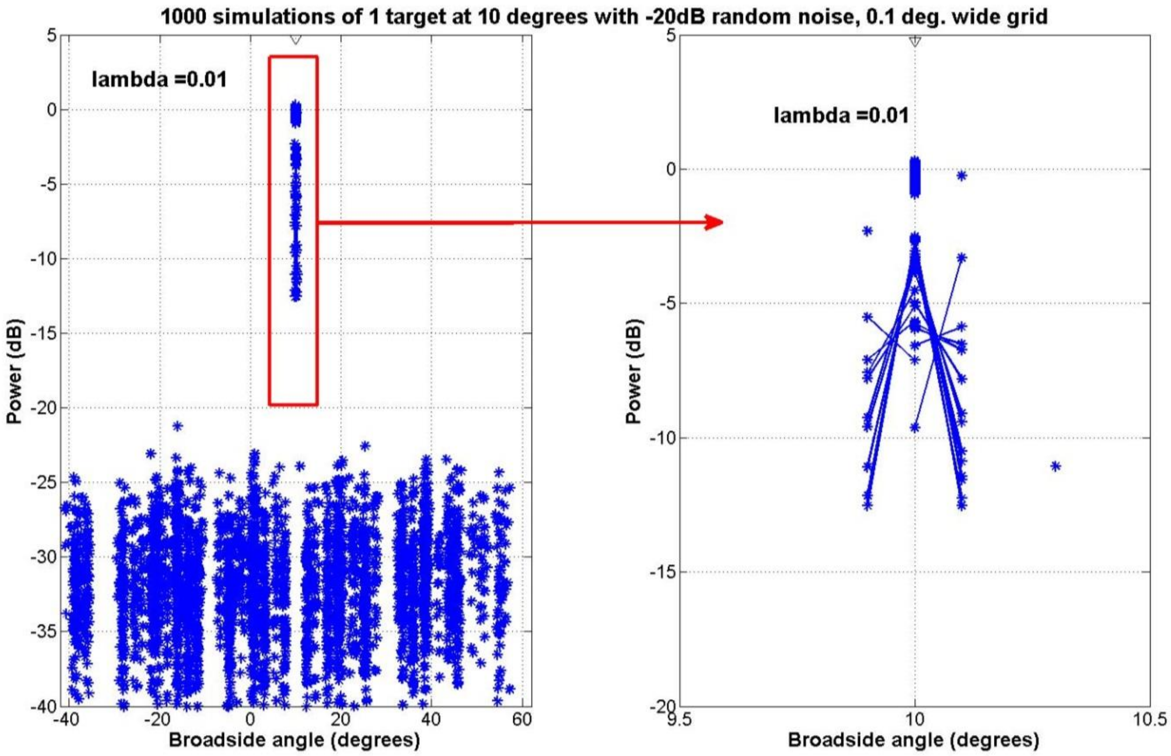

Figure 36: Results for 1000 simulations of a single on grid target on a 0.1 degree grid spacing, with λ=0.01.

The regularization parameter λ plays an important role in the accuracy of the solution since it is a compromise between finding a solution that is as sparse as possible and that has the lowest error as possible. Increasing λ means a “more sparse” solution is sought. As stated in section 3.4.3 the best option is to determine λ for each iteration instead of using the same value for each iteration. Although the GCV have been implemented in the MFOUCSS algorithm the computational load for 1000 simulations increases the computational time enormously. Hence the results have been evaluated by using predefined input values for λ which are used for every iteration. Results of several single runs with different off grid distances, using the GCV technique, where used to find reasonable values for λ. In these runs values for λ between 0.00006 and 0.99 for different iterations within the same run where observed. This states the importance of using a varying λ for each iteration, but it makes it also difficult to pick a “good” constant value for λ. For this thesis values between 0.01 and 0.5 for λ where used. λ=0.01 was chosen since the algorithm states that λ is “Generally close to the noise variance” and λ=0.5 was chosen since it is approximately the average λ found with several different simulations with the GCV technique.

The role of λ is graphically shown in Figure 37. Increasing λ from 0.01 (shown in Figure 36) to 0.5 decreases the noise in the solution and reduces also the spreading and the overall noise level.

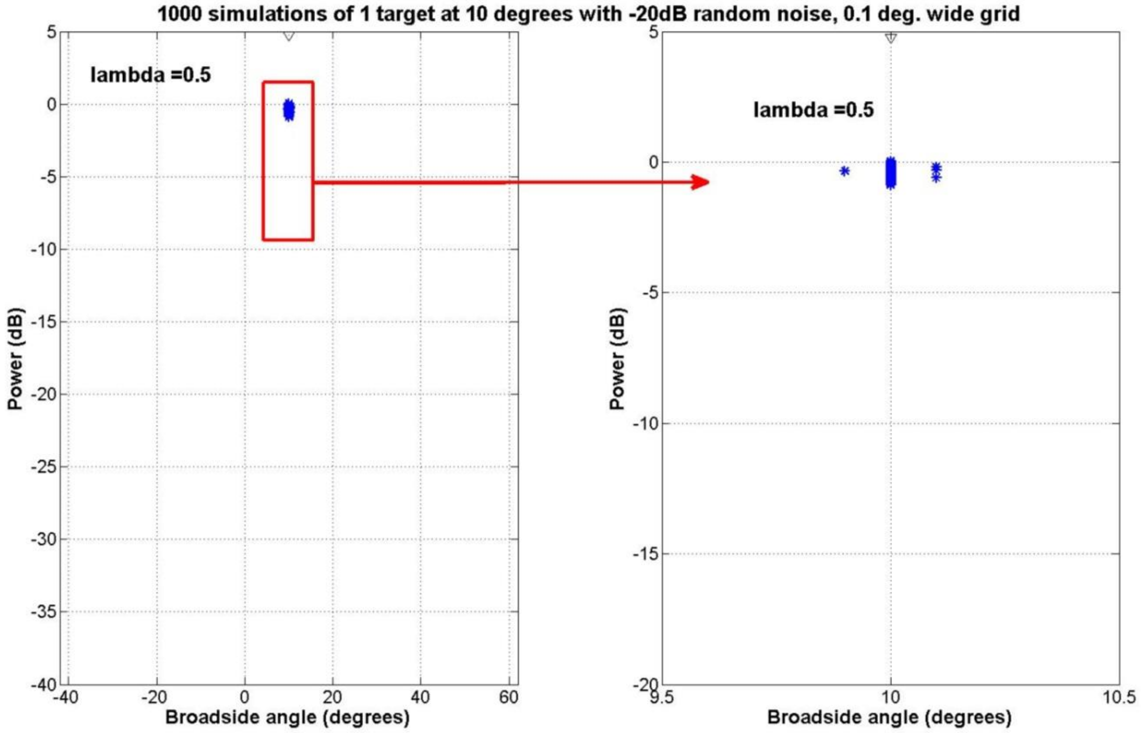

Figure 37: Results for 1000 simulations of a single on grid target on a 0.1 degree grid spacing with lambda=0.5.

One could argue at this point that choosing a value >1 for λ might give very good results. This can be true for a single target, when more targets are present the algorithm will be forced to find a single target solution. This issue is addressed further in the next section regarding the resolution. It is thus important to find a compromise between the grid spacing and λ that minimizes the discretization errors but also gives exact solutions when targets are (almost) on grid. For the -20dB noise case this compromise have been found after several simulation sessions with varying grid spacings and values for λ. The results for the compromise are shown in Figure 38. For a λ of 0.5 the results on a grid spacing of 0.25 degree show no spreading in angle and the noise disappears.

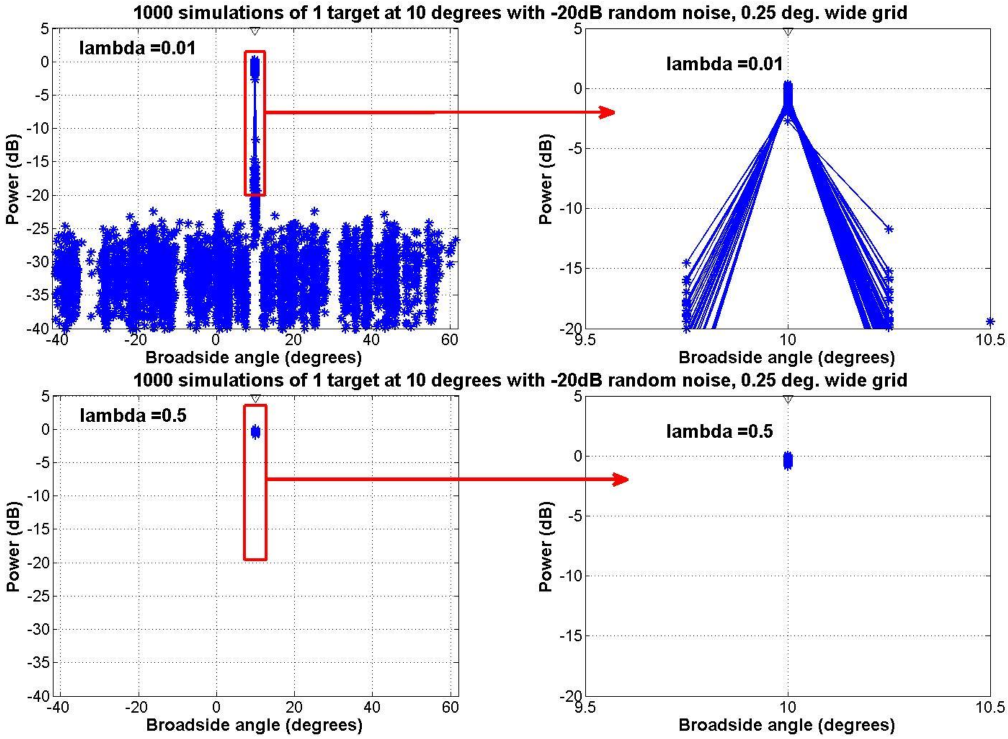

Figure 38: Results for 1000 simulations of a single on grid target on a 0.25 degree grid spacing with lambda=0.01 and 0.5.

When targets are off grid discretization errors can influence the angular accuracy. When the targets are off grid and exactly in between two gridpoints, the spreading of the solution over the 2 adjacent gridlines is inevitable. This automatically decreases the resolution to the grid spacing. When targets are off grid but closer to one gridpoint, it is important for the accuracy that the solution is found at the closest gridpoint. Simulating 1000 runs have shown that this is not always the case. This is shown in Figure 39, where 1000 single target time samples with an off grid DOA have been simulated. Although the target is 0.0625 degree off grid and closer to the gridpoint at 10 degrees, also for every solution, a "spread solution" at 10.25 degrees is found with power levels up to -6dB.

Increasing λ can reduce the spreading of the found solutions for off grid targets as can be seen in Figure 40. With λ=0.5 only 8 of the 1000 simulations have a "spread out solution" at the adjacent gridpoint. This is an improvement to the results of Figure 39 and an increase of the maximal obtainable accuracy. Having an SNR of 20 dB and a grid-cell size of 0.25 degrees, the accuracy is given only by your grid cell size.

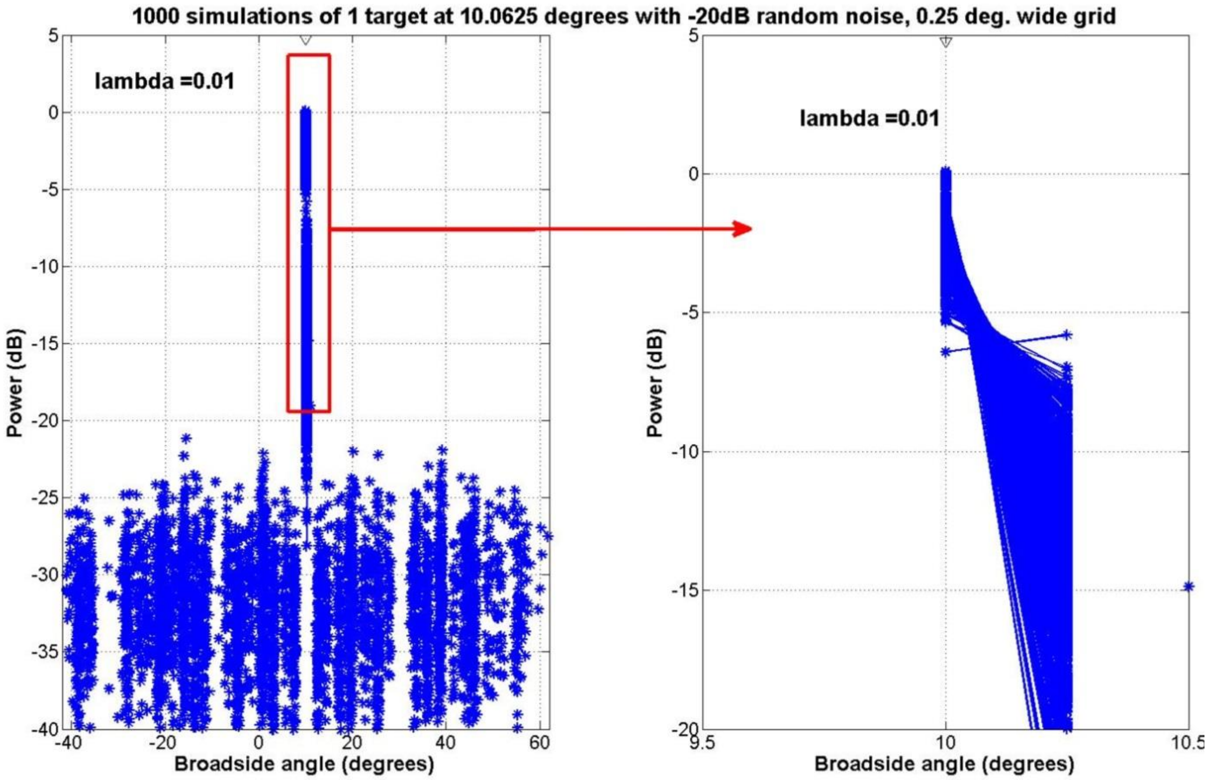

Figure 39: Results of discretization errors on 1000 simulations on a 0.25 degree spaced grid.

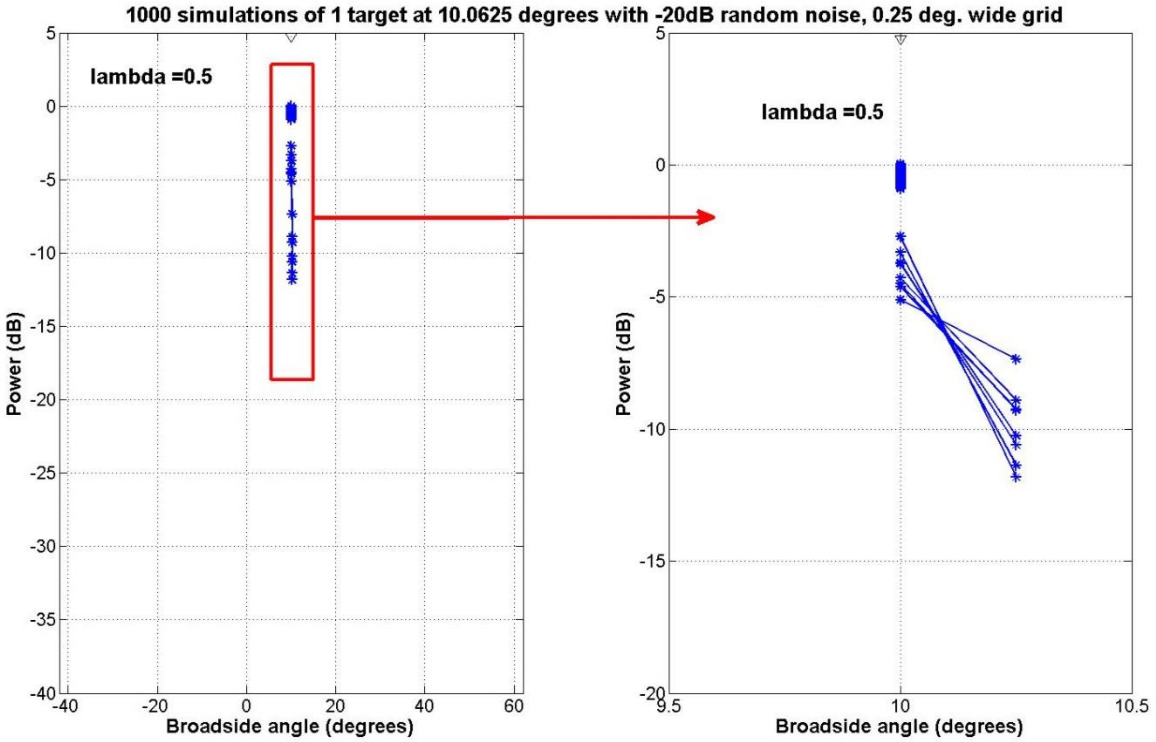

Figure 40: Results of discretization errors on 1000 simulations on a 0.25 degree spaced grid and λ=0.5.

4.4.2 Discussion and Conclusions regarding Accuracy

The accuracy is determined by a compromise between the grid spacing and the regularization parameter λ, which minimizes the discretization errors but also gives exact solutions when targets are (almost) on grid. Besides that increasing the regularization parameter also decreases the level of the overall noise found in the solution. The best option in terms of finding the most accurate solution is to determine the regularization parameter before each iteration with for instance the GCV technique. Since this comes at the cost of an increased computational load this is not simulated for 1000 runs and recommended as a topic for further research.

In the description of the algorithm it is written that a value close to the noise variance for λ generally gives good results. Hence this value was chosen as a lower bound for λ. It gave however solutions which were spread out to adjacent gridlines with power levels up to -6dB. Simulations showed that a maximal λ of 0.5 resulted in exact solutions for on grid targets and a significant decrease in spread out solutions for off grid targets. The spreading of the solution for the 0.1 degrees cell size is caused by the noise and determines the accuracy in the angle estimation. If you choose a grid cell of 0.25 degrees your accuracy will be given by the grid cell size, with an SNR of 20dB. For on grid targets it can be said that the accuracy is ≤ 0.25 degrees. This is roughly a factor 3 more than the theoretical limit/Cramer Rao Bound of 0.07 degrees of the fully dense array.

When the targets are off grid and exactly in between two gridpoints, the spreading of the solution over the 2 adjacent gridlines is inevitable. Simulating 1000 runs with a grid spacing of 0.25 degrees and an 0.0625 degrees off grid target showed that when a value of λ=0.5 is chosen the accuracy is not decreased and still ≤0.25 degrees.

4.4.3 Resolution: two targets

An important question that arises when two targets are present at the same time is at what angular distance they can be distinguished from each other. When the combined signal of two targets is received and used as input for the MFOCUSS algorithm and the targets are too close they merge into a single target and cannot be distinguished anymore. Obviously the size of the grid plays a role in this. When the targets are within one grid spacing they obviously cannot be distinguished since they are merged by the discretization. On the other side, the finer the grid gets, the more spreading in the solutions can occur, making the targets merge into a single cluster. When the limit of being able to distinguish between two clusters is reached, this is assumed the maximal resolution possible.

For conventional phased arrays it is generally accepted that two equal targets can be resolved in angle when they are separated at least eight-tenths of a beamwidth. Since the beamwidth, according to equation (7), of the fully dense array is 1.79°. Using equation (15), the best achievable angular resolution for an SNR of 20dB will approximately 1.04° (Still provided that the SNR is large enough for detection). For our CS sparse array we want to approach this value as much as possible in the first place. For this thesis an SNR of 20dB is assumed to be large enough for detection, and thus with a noise level of -20dB we want to verify by simulations that the minimal angular resolution of 1.04° can be approached.

Simulations using a 1 degree spaced grid showed that off grid targets with a 2 degree separation cannot be distinguished. This is shown in Figure 41 where the results are shown of 1000 simulations for targets with DOAs 10.5 and 12.5 degrees within -20dB noise. The results show a single “cluster” of detections, hence for a single detection it is not possible to certainly say that two targets are arriving instead of one.

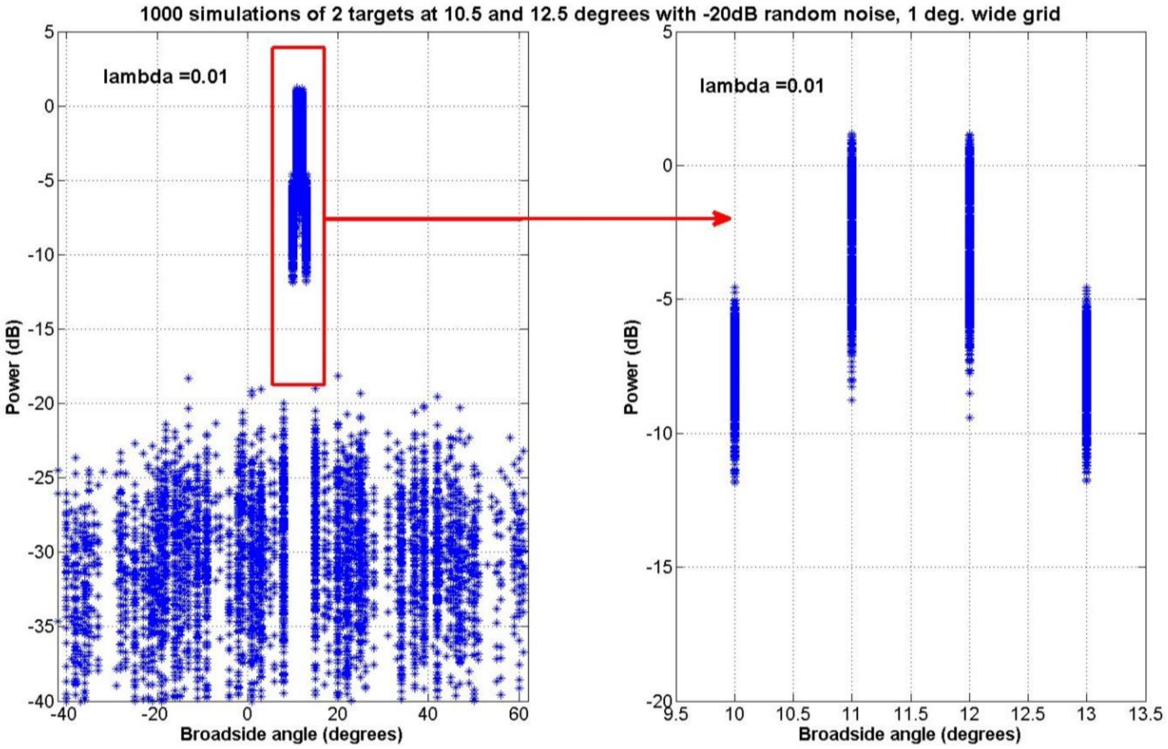

Figure 41: Result from 1000 simulations for 2 off grid targets on a 1 degree spaced grid.

Obviously the targets also merge since the grid spacing is 1 degree, and the targets are exactly in the middle of two gridpoints. For the accuracy a grid spacing smaller than 0.25 degrees resulted in spreading when $\lambda < 0.5$, even for on grid situations. Since we want to approach the resolution of the equivalent dense array of 1.27 degrees, simulations have been done for targets spaced at least 1.25 degrees/5 gridpoints. The targets have been given a random DOA between 10.250-10.375 and 11.625-11.750 respectively. Hence they are at least separated 1.25 degrees and at the same time the (random varying) off grid effect is taken into account. The results for 1000 simulations with λ equal to the noise variance are shown in Figure 42. Clearly two clusters of targets can be seen since there are no solutions found on the gridpoints at 10.75, 11.0 and 11.25 degrees.

The results in Figure 42 have λ chosen as the noise variance. Increasing λ to 0.5 creates a more sparse solution as shown in Figure 43. For a single target choosing λ=0.5 resulted in spreading only on the adjacent gridpoints (Figure 40). For two targets and λ=0.5 the spreading now goes over three gridpoints (Figure 43). For λ=0.01 this is also the case but there is also more spreading in amplitude in the solutions as can be seen in Figure 42.

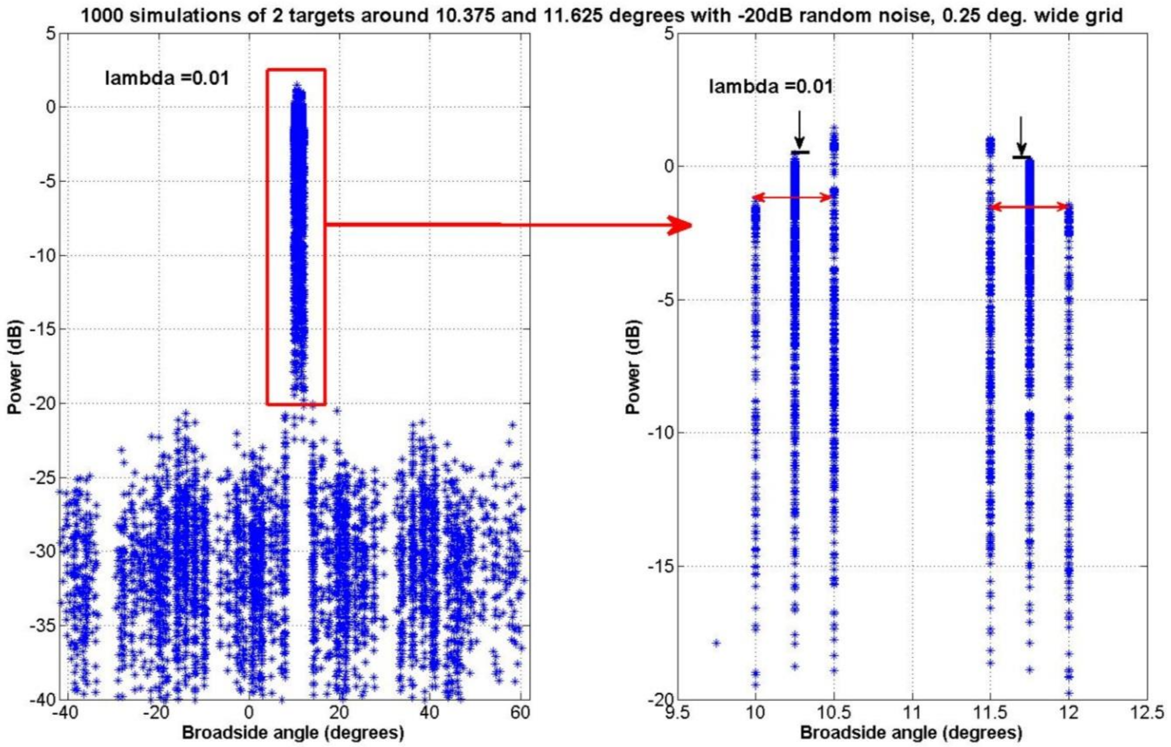

Figure 42: Results from 1000 simulations for 2 off grid targets on a 0.25 degree spaced grid with λ=0.01. The DOAs are randomly chosen between 10.250 and 10.375 and 11.625 and 11.750 degrees. The red double arrows indicate the spread in the solutions, the black arrow the angle span of true DOAs.

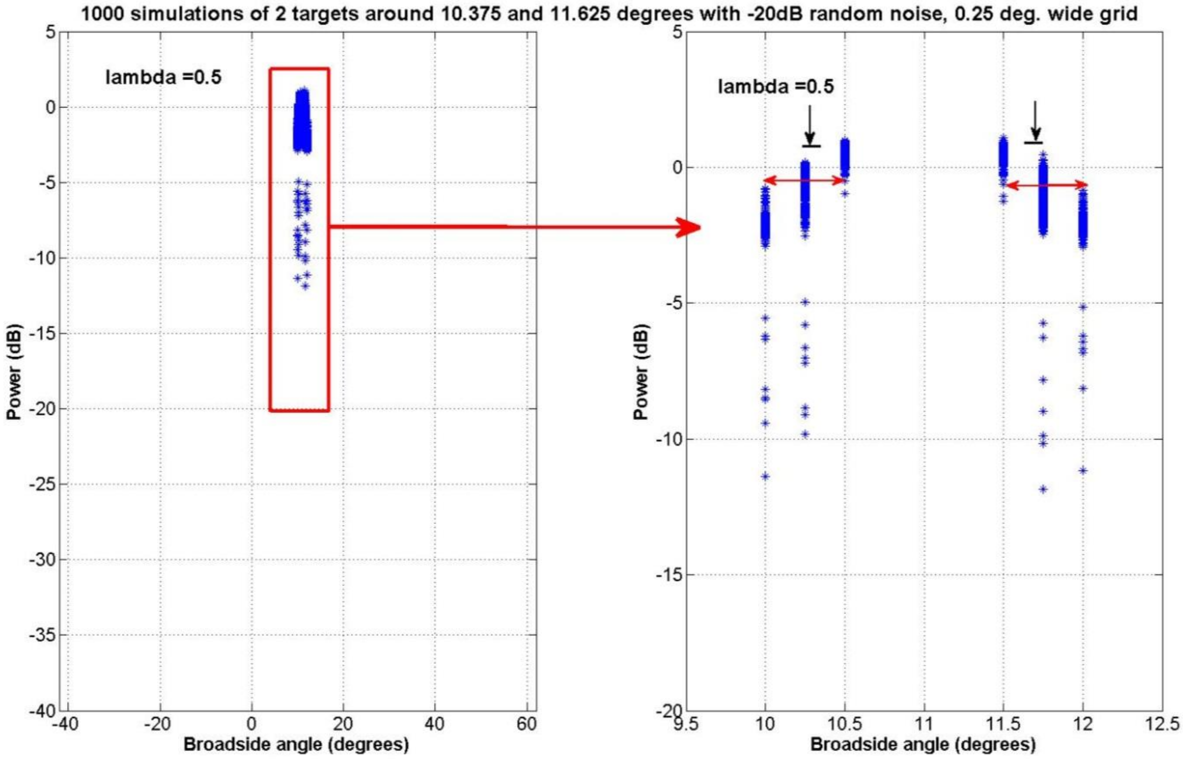

Figure 43: Results from 1000 simulations for 2 off grid targets on a 0.25 degree spaced grid with λ=0.5. The DOAs are randomly chosen between 10.250 and 10.375 and 11.625 and 11.750 degrees. The red double arrows indicate the spread in the solutions, the black arrow the angle span of true DOAs.

It is not possible to keep increasing λ to find a more sparse solution for each of the two targets. When more than one target is present the algorithm will be forced to find a single target solution when λ is chosen to high. To show this effect the results for (a high value of) λ=10 are shown in Figure 44 for which the same 1000 random simulations are used as in Figure 43. The result is that 143 out of 1000 simulations give a solution which represents a single target at 11 degrees, which is in the middle of the two actual arriving targets. The found solution is thus representing more a single target instead of two targets at different angles.

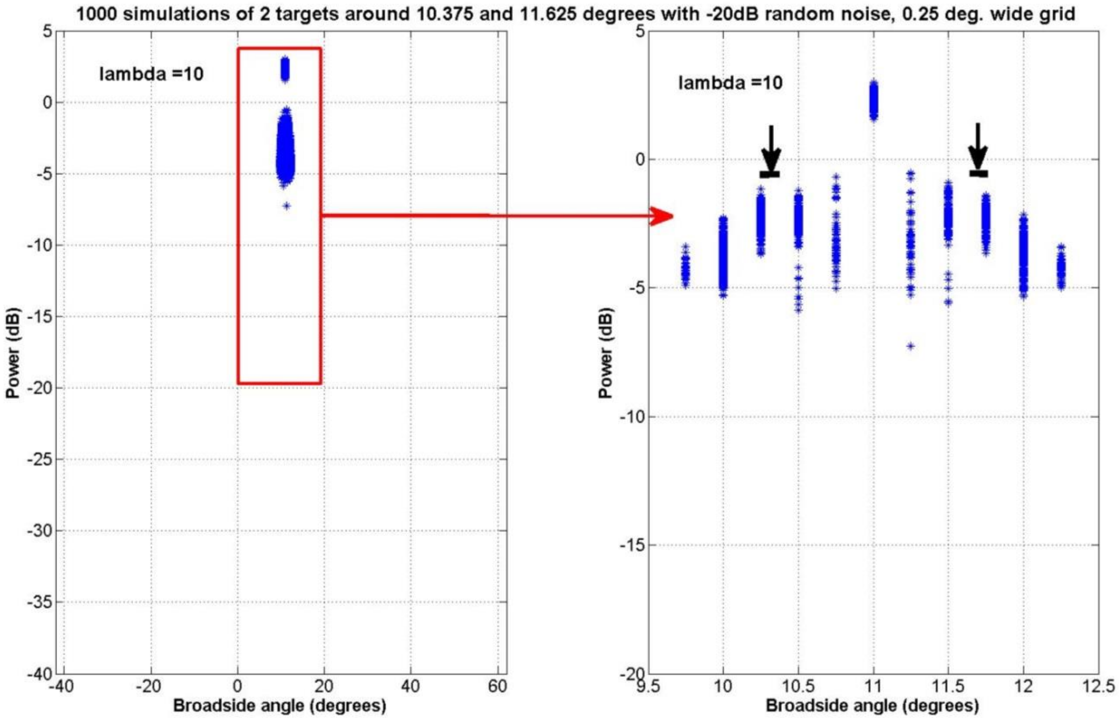

Figure 44: Results from 1000 simulations for 2 off grid targets on a 0.25 degree spaced grid with λ=10. The DOAs are randomly chosen between 10.250 and 10.375 and 11.625 and 11.750 degrees. The black arrows indicate the bracket in which the true DOAs are.

Now that simulations have shown that there are three empty gridpoints between the two targets, the question arises if the CS array might be able to achieve even a better result. Therefore 1000 simulations with two targets with a random DOA between 10.375-10.5 and 11.5-11.625 respectively, being at least 1 degree separated, have been simulated for various λ values. For none of the 1000 simulations, solutions in between the two targets (at the 11 degree gridpoint) where observed. For λ=0.25, the solutions found in between the two targets where minimal.

The results for λ=0.25, are shown in Figure 45. On the gridpoint in the middle of the two targets, at 11 degrees, no solutions are found. On the gridpoints at 10.75 and 11.25 are 2 and 3 solutions found respectively. The number of found solutions on the 10.75 and 11.25 gridpoints both increased when λ was decreased ánd increased to 0.01 and 0.5 respectively. This means that λ-0.25 is the optimal value to choose in this case as it results in the most solutions not being in the middle of the two actual DOAs. The spread of the solutions is however increased from 3 to 4 gridpoints.

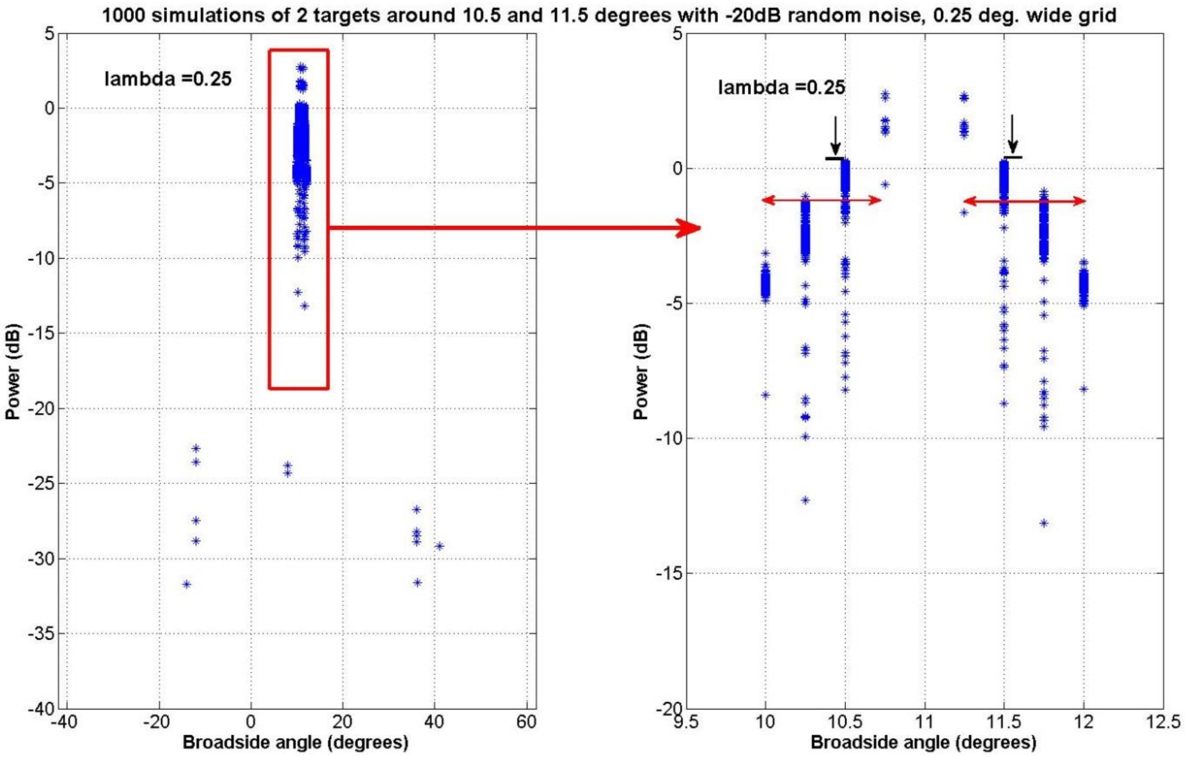

Figure 45: Results from 1000 simulations for 2 off grid targets on a 0.25 degree spaced grid with λ=0.25. The DOAs are randomly chosen between 10.375 and 10.5 and 11.5 and 11.625 degrees. The red double arrows indicate the spread in the solutions, the black arrow the angle span of true DOAs.

To test the minimal separation of 1 degree extensively, 1000 simulations with a (different) random noise of -20dB have been done for targets with exactly 1 degree separation. This is shown in Figure 46 whereas 1000 random noise situations, with targets arriving from the on grid DOAs 10.5 and 11.5 degrees, have been simulated with λ=0.5. Since this higher value for λ forces the algorithm to find sparser solutions, 3 out of 1000 simulations gave a "single target" solution. They are indicated by the red arrow in Figure 46. These "single target" solutions have been observed for values of λ>0.25 for a 1000 simulations with random noise at -20dB. In case λ is set at 0.25 there are no single target solutions anymore and the gridpoint at 11 degrees is again free of solutions above -20dB.

To illustrate the effect of noise on the solution, the noise was increased up to the point where the algorithm started to create "single target" solutions with λ=0.25 and the same DOAs as used in Figure 46. Already at a noise level of -19 dB the first "single target" solutions occurred for 1000 simulations.

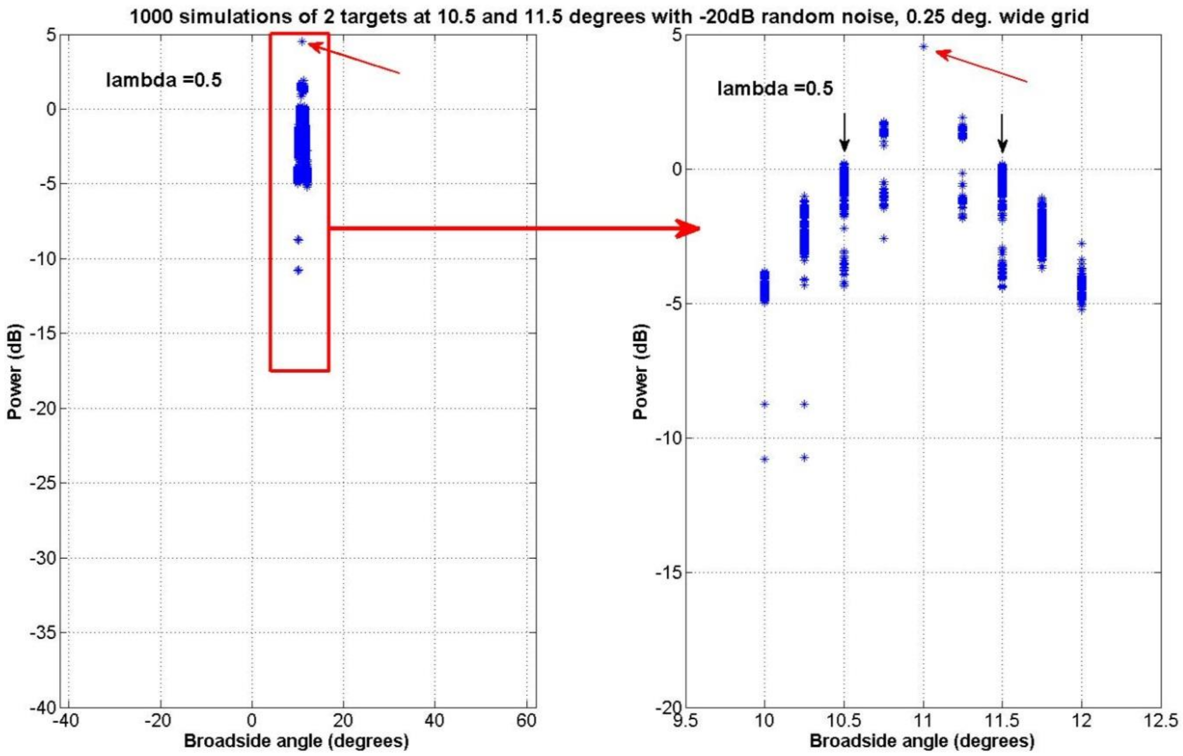

Figure 46: Results from 1000 simulations for 2 on grid targets, separated 1 degree, on a 0.25 degree spaced grid with λ=0.5. The black arrows indicate the true DOAs.

4.4.4 Discussion and Analysis of the Resolution

Taking into account the off grid effect and an SNR of 20dB, it was shown that targets at least separated 1.25 degrees clearly resulted into two clusters of solutions when 1000 runs where simulated. There have been no solutions found on the three gridpoints in between the two targets. This makes it highly plausible that a cluster and detection algorithm can distinguish the two targets from each other. The resolution of the equivalent fully dense array of 1.26 degrees can hence be achieved with the sparse array with CS when the SNR is at least 20dB.

Increasing λ to improve the accuracy was advantageous. For the resolution it can be better to decrease λ as two targets are moving closer to each other. This to avoid enforcing the algorithm in finding "single target" solutions while actually two targets are present. For two targets spaced ≤ 1.25 degrees a $\lambda \geq 1$ will enforce the algorithm to find a solution for a single target instead one for two targets since this is more sparse. Hence one has to be careful with increasing λ. This indicates again the importance of finding a good compromise and thus λ between the sparsity and the accuracy of the solution. Therefore it is advised to determine the optimal λ before for each iteration. This can be done with for instance the GCV technique, at the cost of a significant increase in computational load. More research to efficiently find an optimal λ before for each iteration is recommended.

Simulations showed that choosing λ no larger than 0.25 resulted in trustworthy results, containing two target solutions for targets spaced at least 1 degree within -20dB noise. In this situation the gridpoint in the middle of the two targets did not contain any solution for 1000 simulations. Thus, although marginal, it is plausible that targets can be distinguished by a detection algorithm when they arrive with a separation of 1 degree when the SNR= 20dB. The fact that only one gridpoint in between the two targets contains no solutions emphasizes that a resolution of 1 degree, with a carefully chosen λ, is the maximal achievable resolution for two equal targets with SNR=20dB. The achieved resolution is in accordance with the theoretical Cramér-Rao bound, under the condition that the right value for λ is chosen. Choosing the right value for λ does depend on the situation, but in general experimental results of λ=0.25 delivered good results.

The maximal resolution depends also highly on the development of a good cluster and detection algorithm. The cluster algorithm must be able to combine and cluster the obtained solutions to represent the correct number of targets present. The detection algorithms must be able to detect the clusters above the noise level, eventual using multiple measurements. The algorithms must be able to do this without any a priori information. Creating or implementing such an algorithm is beyond the scope of this thesis and a recent research topic in the field of CS Radar [31].

It is also noted that the obtained results are at a scanangle of 10 degrees. Since scanning increases the beamwidth with a factor $\frac{1}{\cos\theta}$ (see section 2.2.3) the results for the accuracy and resolution are expected to be a factor $\frac{1}{\cos\theta}$ worse when the targets arrive at an angle θ off broadsight.

5 Conclusions & Recommendations

This chapter presents the conclusions forthcoming from the research done for this thesis. The conclusions are divided into two parts. The first part contains conclusions related to the simulations that have been done and thus to the implementation of CS for signal reconstruction using sparse arrays. This part will also answer the question if it is possible to reconstruct the received signal of a sparse or faulty array with CS and, using this reconstruction, to estimate the DOA of incoming targets using the reconstruction. Conclusions regarding the required SNR, maximal number of targets that can be reconstructed at the same time and the accuracy and resolution are also presented in the first section. The second part will address the conclusions related to the use of the MFOCUSS algorithm and its implementation in Matlab. The section ends with recommendations for further research.

5.1 Conclusions on the use of CS for signal reconstruction

The conclusions are valid under the following assumptions:

- The simulations that have been done with the ULA consisting of 64 elements, of which only 16 random elements are working. The outermost elements of the array are always working.
- Targets are within the sector from -60 to 60 degrees with relation to the boresight direction;
- The discretization grid is defined from -90 to 90 degrees with relation to the boresight direction;
- The reduce in gain due to less transmitting elements is not taken into account.

The research done under these assumptions resulted in the following conclusions regarding the use of compressive sensing for DOA estimation:

Without noise:

The relation between the number of measurements *M,* needed to reconstruct *K* sparse signals is verified for the noiseless case: *K* equal strength targets on the grid without noise can be recovered exactly, as long as there are *M*=*K* log(*N*) random working elements in the sparse array.

With Noise:

For a single target with a SNR ≤ 30dB the reconstructed signals show noisy effects due to off grid effects.

For an SNR≥30dB, when *K* targets are off grid, with random strengths within -10 and 0dB, the number of targets that can be correctly reconstructed is in the order of $\frac{M}{\log(N)} - 1$;

Using multiple time snapshots in noise

The noisy effects that come from off grid effects and received noise, can be decreased significantly by using more time samples.

When 10 time samples are used the condition $K = \frac{M}{\log(N)}$ is valid again for SNR ≥10dB. When 3 time samples are used the condition is valid again for SNR ≥20dB. This is only valid under the assumption that the used time samples have the same sparsity profile.

Using multiple time snapshots makes it very plausible that CS (using MFOCUSS) can be used in a real life system to maintain DOA estimation performance when a large part of the elements in a dense array fail or a random sparse array is used.

Conclusions on the accuracy and resolution

The accuracy depends on the noise, the grid spacing and the regularization parameter λ. For a noise level of -20dB, a maximal value for the regularization parameter of *λ=0.5* and a grid spacing of 0.25 degrees the accuracy of the simulated system is 0.25 degrees.

Choosing a grid spacing smaller than 0.25 degrees can result in a decreasing accuracy since the solutions tend to spread over several gridpoints, even is the target is arriving exactly on grid. This is due to the increased coherence of the sensing matrix, for a finer grid. Thus it is advisable not to choose a grid spacing smaller than 0.25 degrees when the resolution of the fully dense array is equal to $\frac{\lambda_w}{D} = 1.79°$.

Simulations have shown that a resolution in the order of the resolution of the equivalent fully dense array of 1.04 degrees is feasible. With an intelligent detection algorithm the resolution might even be increased to the theoretical limit of 1.04 degrees.

The maximal obtainable resolution does not only depend on the correctness of the found solution but also on the ability to interpret the found solution. This involves the use of a cluster and detection algorithm that can determine if there is one target or that there are two closely spaced targets, without any a priori information.

Values of λ=0.25 on a 0.25 degree spaced grid achieved the best resolution. For the resolution it can be better to decrease λ as two targets are moving closer to each other. This to avoid enforcing the algorithm in finding “single target” solutions while two targets are present

5.2 Conclusions on the use of the MFOCUSS algorithm

For a single target noise gives rise to noise in the reconstruction, but the input noise is never increased by the algorithm. Thus the noise in the reconstruction is never higher than the input noise;

When targets are not exactly on the discretization grid, discretization errors or discretization noise appears in the solutions. A way to decrease the reconstruction noise is to use a finer grid, thus increasing the probability that targets are on the grid or at least closer to the grid. This however increases computational complexity and thus the time to find a solution. Using a very fine grid increases the probability of numerical instability, making failing of the algorithm more likely. Simulations have shown that gridspacings smaller than 0.25 degrees resulted in reconstruction errors, which could be (partially) due to numerical instabilities.

The best choice for the regularization parameter λ that is needed for the MFOCUSS algorithm in the presence of noise is not in the order of the noise algorithm, as is stated in the description of the algorithm. Generally better solutions, with less noise, are found when λ is chosen higher than the noise variance. Improving solutions by changing λ to its optimal value works also in the presence of noise and with discretization errors.

The best option is to determine λ for every iteration instead of using the same predefined value for every iteration. Especially for off grid targets in the presence of noise this results in more accurate and less spread out solutions. Determining λ with the Generalized Cross Validation technique can be implemented in the algorithm. This gives the advantage of faster convergence for off grid targets in noise and not having to choose a value for λ as in input parameter. The downside is however the increase in computational load. This increase can be of such a magnitude that real time processing might not be possible anymore. One should also carefully choose the bounds of λ when it is determined using GVC. When values above 0.25 are possible for GVC to use, it is highly likely the resolution decreases. This is due to the fact that high values for λ force the algorithm to find a “single target solution”, instead of a “two closely spaced target” solution.

5.3 Recommendations

The following topics are recommended for further research:

- Change the antenna from a 1 dimensional line array antenna to a 2 dimensional planar array antenna and research the possibilities of using the CS reconstruction method. 2-dimensional planar array antennas are used in practical radar systems, but using 2 dimensional planar arrays could complicate the problem.
- Study ways to decrease the computational load of the GCV technique, or find computational more efficient techniques to determine the regularization parameter λ before each iteration. There are clear advantages for determining λ before every iteration, but during this thesis the increase in computation load was to high obtain results that could represent real time processing. Besides that studying the effects of varying the regularization parameter λ for more than 2 arriving targets is recommended. This thesis focused on the resolution between two targets and the effects of λ for the resolution. It is not certain if the same effects of changing λ and the grid spacing occur when more than 2 targets arrive.
- Study the effects of antenna model errors (element position errors, mutual coupling effects etc.) on the CS array. This is a step closer to real life implementation.
- Study the possibility of where to implement the CS processing in the total processing chain. This can make clear if the use of MMV is feasible and if Doppler filtering can give additional sparsity within different Doppler bins.
- Developing and implementing a cluster and detection algorithm that can be used in the processing chain. This makes the use of a CS phased array comparable to a “traditional” matched filter phased array.
- During the thesis a 3dB loss in reconstructions for off grid signals was observed, but the cause was not found. It is highly likely that it is because of a wrong normalization that occurs for off grid signals within the algorithm. Recommended is to find the cause of this loss off energy to present results at their true values.

References

[1] V. Krishnaveni, T. Kesavamurthy en B. Aparna, „Beamforming for Direction-of-Arrival Estimation - A survey,” *International Journal of Computer Applications,* vol. 61, nr. 11, pp. 4-11, 2013.

[2] C. A. Balanis, Antenna Theory (Third edition), Hoboken, New Jersey: John Wiley & Sons Inc., 2005.

[3] M. I. Skolnik, Introduction to RADAR systems, NY: McGraw-Hill Book Companies, 2001.

[4] E. Stolp, „Resolution and Accuracy Analysis,” Thales Internal report, 2015.

[5] B. Hayes, „THe Best Bits,” *American Scientist,* nr. 97, pp. 275-280, 2009.

[6] E. Candès, J. Romberg en T. Tao, „Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information,” *IEEE Transactions on Information Theory,* nr. vol. 52, no. 2, pp. 489-509, Feb. 2006.

[7] D. Donoho, „Compressed Sensing,” *IEEE Transactions on Information Theory,* nr. vol 52, no. 4, pp. 1289-1306, Apr. 2006.

[8] V. Cevher, A. C. Gurbuz, J. H. McClellan en R. Chellappa, „Compressive Wireless Arrays for Bearing Estimation,” *IEEE International Conference on Acoustics, Speech and Signal Processing,* pp. 2497 - 2500 , 2008.

[9] A. C. Gurbuz, J. H. McClellan en V. Cevher, „A Compressive beamforming method,” *IEEE International Conference on Acoustics, Speech and Signal Processing, 2008.,* pp. 2617 - 2620, 2008.

[10] E. de Jong en R. Pribic, „Sparse signal processing on estimation grid with constant informatiion distance applied in radar,” *EURASIP Journal on Advances in Signal Processing,* 2014.

[11] R. Pribic en I. Kyriakides, „Design of sparse-signal processing in radar systems,” *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pp. 5008 - 5011, 2014.

[12] J. H. Ender, „On Compressive sensing applied to radar,” *Elsevier Signal Processing,* Vols. %1 van %2Volume 90, issue 5, pp. 1402-1414, 2010.

[13] Y. Wang, G. Leus en A. Pandharipande, „Direction Estimation Using Compressive Sampling Array Processing,” *Statistical Signal Processing, 2009. SSP '09. IEEE/SP 15th Workshop on,* pp. 626-629, 2009.

[14] S. Wei, X. Zhang, J. Shi en K. Liao, „Sparse array microwave 3-d imaging: compressed sensing recovery and experimental study,” *Progress In Electromagnetics Research,* vol. Vol. 135, pp. 161-

181, 2013.

[15] Y. Han en J. Wang, „Adaptive Beamforming Based on Compressed Sensing with Smoothed l0 Norm,” *International Journal of Antennas and Propagation,* 2015.

[16] R. G. Baraniuk, „Compressive Sensing,” *IEEE Signal Processing Magazine,* pp. 118-124, Jul. 2007.

[17] E. Candès en M. Waking, „An Introduction To Compressive Sampling,” *IEE SIgnal Processing Magazine,* pp. 21-30, Mar. 2008.

[18] E. Candes en T. Tao, „Decoding by linear programming,” *IEEE Trans. Inform. Theory,* vol. 51, nr. 12, pp. 4203-4215, 2005.

[19] Z. Zhang, „Codes,” University of California DSP LAB, [Online]. Available: http://dsp.ucsd.edu/~zhilin/Software.html. [Geopend 03 2016].

[20] Rice University, „Compressive Sensing Resources,” Rice University, [Online]. Available: http://dsp.rice.edu/cs. [Geopend 03 2016].

[21] S. F. Cotter, B. D. Rao, K. Engan en K. Kreutz-Delgado, „Sparse Solutions to Linear Inverse Problems With Multiple Measurement Vectors,” *IEEE Transactions on signal processing,* vol. 53, nr. 7, pp. 2477-2488, 2005.

[22] I. F. Gorodnitsky en B. D. Rao, „Sparse Signal Reconsruction from Limited Data Using FOCUSS: A Re-weighted Minimum Norm Algorithm,” *IEEE Transactions On Signal Processing,* vol. 45, nr. 3, pp. 600-616, 1997.

[23] G. Tang, B. N. Bhaskar, P. Shah en B. Recht, „Compressive Sensing off the grid,” *IEEE Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on,* pp. 778-785, 2012.

[24] X. Han, H. Zhang en G. Li, „Fast Algorithms for Sparse Recovery with Perturbed Dictionary,” *arXiv,* 2012.

[25] J. C. Ye, S. Tak, Y. Han en H. W. Park, „Projection reconstruction MR Imaging using FOCUSS,” *Magnetic Resonance in Medicine,* nr. 57, pp. 764-775, 2007.

[26] I. F. Gorodnitsky, J. S. George en B. D. Rao, „Neuromagnetic source imaging with FOCUSS: a recursive weighted minimum norm algorithm,” *Elsevier Electroencephalography and clinical Neurophysiology,* vol. 95, pp. 231-251, 1995.

[27] B. D. Rao, K. Engan, S. F. Cotter, J. Palmer en K. Kreutz-Delgado, „Subset Selection in Noise Based on Diversity Meausure Minimization,” *IEEE Transactions on Signal Processing,* vol. 51, nr. 3, pp. 760 - 770, 2003.

[28] E. Ghaoui, „Matrix Norms,” University of California, [Online]. Available:

https://inst.eecs.berkeley.edu/~ee127a/book/login/l_mats_norms.html. [Geopend 03 2016].

[29] R. Zdunek en A. Cichocki, „Improved M-FOCUSS algorithm with overlapping blocks for locally smooth sparse signals,” *IEEE Transactions on Signal Processing,* vol. 56, nr. 10, pp. 4752-4761, 2008.

[30] G. H. Golub, M. Heath en G. Wahba, „Generalized Cross-Validation as a Method for Choosing a Good Ridge Parameter,” *Technometrics,* vol. 21, nr. 2, pp. 215-223, 1979.

[31] Fraunhofer FHR, „CoSeRa 2016, International Workshop on Compressed Sensing Theory and its Applications to Radar, Sonar, and Remote Sensing,” [Online]. Available: http://workshops.fhr.fraunhofer.de/cosera/. [Geopend March 2016].

[32] I. F. Gorodnitsky en B. D. Rao, „A Recursive Weighted Minimum Norm Algorithm: Analysis and Applications,” *Proc. Int. Conf. on Acoustic, Speech and Sign. Processing,* vol. III, pp. 456-459, 1993.

Appendix A: Regularized MFOCUSS with GCV m-file

```
function [X, gamma_ind, gamma_est, count, lambda_history, dmu_history2] =
mfocuss_lhat(A, Y, varargin)
% MFOCUSS algorithm for the MMV model
%    Mainbody from http://dsp.ucsd.edu/~zhilin/MFOCUSS.m
% ============================== INPUTS ==============================
%   A              : N X M dictionary matrix
%
%   Y              : N X L measurement matrix, i.e. Y = Phi * X + V.
%
%  'p'             : p-norm. p lies in [0,1].  Default value: p = 0.8
%
%  'PRUNE_GAMMA'   : Threshold for prunning small gamma_i.
%                    In noisy cases, you can set PRUNE_GAMMA = 1e-3 or 1e-4.
%                    In strongly noisy cases (SNR<=5 dB), suggest to set
PRUNE_GAMMA = 0.01;
%                    Default value: MIN_GAMMA = 1e-4.
%
%  'MAX_ITERS'     : Maximum number of iterations.
%                    Default value: MAX_ITERS = 800
%
%  'EPSILON'       : Threshold to stop the whole algorithm.
%                    Default value: EPSILON = 1e-8
%
%  'PRINT'         : Display flag. If PRINT = 1: show output; If PRINT = 0:
supress output
%                    Default value: PRINT = 0
%
% ==============================  OUTPUTS ==============================
%   X             : Estimated solution matrix(size: M X L)
%   gamma_ind     : Indexes of nonzero gamma_i
%   gamma_est     : Final values of all the gamma_i (including zeros). An M
X 1 vector
%   count         : number of iterations used
%
% ============== Examples of Commands ===============
% [Example 1]
%   lambda = 1e-3;
%   [X,gamma_ind,gamma_est,count] ...
%       = MFOCUSS(Phi,Y,lambda,'p',0.8,'prune_gamma',1e-
4,'max_iters',500,'epsilon',1e-8,'print',0);
%
% [Example 2]
%      lambda = 1e-5;
%      [X,gamma_ind,gamma_est,count] = MFOCUSS(Phi,Y, lambda);
%
% ==============================  Reference ==============================
%   [1] Cotter, S.F.;   Rao, B.D.;   Kjersti Engan;   Kreutz-Delgado, K.;
%       Sparse solutions to linear inverse problems with multiple
measurement vectors
%   [2] Zdunek, R.; Cichocki A.;
%       Improved M-FOCUSS Algorithm with overlapping blocks for locally
%       smooth sparse signals.
% ==============================  Author ==============================
%   Zhilin Zhang (z4zhang@ucsd.edu)
%   Mainbody was written by David Wipf
%   Renaming parameters cf ref [1] and adding the Generalized Cross
%   Valitation Technique to determine lambda cf ref [2] by Michiel van Tent
%   Beking
% % ==============================  Version ==============================
%   1.0 (04/03/2016)
%

%INITIALIZATION FOR THE UPDATE LAMBDA PART
%Mel=16; %define number of working elements for the golden section search
fgcv = @GCV %define funciton handle for GS search
lambda_min=10^-10; %lower boundry value for lambda
lambda_max=0.99;   %upper boundry value for lambda

% Dimension of the Problem
[N M] = size(A);
[N L] = size(Y);

% Default Control Parameters
PRUNE_GAMMA = 1e-4;        % threshold for prunning small gamma_i
p           = 0.8;         % p-norm
EPSILON     = 1e-8;        % threshold for stopping iteration.
MAX_ITERS   = 800;         % maximum iterations
PRINT       = 0;           % not show progress information

% get input argument values
if(mod(length(varargin),2)==1)
    error('Optional parameters should always go by pairs\n');
else
    for i=1:2:(length(varargin)-1)
        switch lower(varargin{i})
            case 'p'
                p = varargin{i+1};
            case 'prune_gamma'
                PRUNE_GAMMA = varargin{i+1};
            case 'epsilon'
                EPSILON = varargin{i+1};
            case 'print'
                PRINT = varargin{i+1};
            case 'max_iters'
                MAX_ITERS = varargin{i+1};
            otherwise
                error(['Unrecognized parameter: ''' varargin{i} '''']);
        end
    end
end

if (PRINT) fprintf('\nRunning M-FOCUSS for the MMV Problem...\n'); end


% Initializations
Wk = ones(M,1);         % initialization of gamma_i
keep_list = [1:M]';        % record the index of nonzero gamma_i
m = length(keep_list);     % number of nonzero gamma_i
Xk = zeros(M,L);           % initialization of the solution matrix
count = 0;                 % record iterations


% Learning loop
while (1)

    % =========== Prune weights as their hyperparameters go to zero
===========
    if (min(Wk) < PRUNE_GAMMA )
        index = find(Wk > PRUNE_GAMMA); %Find the columns in gamma which
are <PRUNE_GAMMA
        Wk = Wk(index);           %only columns with values > PRUNE_GAMMMA
remain
        A = A(:,index);            % corresponding columns in Phi
        keep_list = keep_list(index);   %store the columns that remain
after removing columns with values <PRUNE_GAMMA
        m = length(Wk);

        if (m == 0)   break;  end;
    end;


    % ====== Compute new weights ======
    Wk = repmat(sqrt(Wk)',N,1);   %=Wk+1 combined with taking the sqrt of
abs(sum(Xk[i])=Ck[i]^2)
    %Wk = diag(sqrt(Wk));   %=Wk+1 combined with taking the sqrt of
abs(sum(Xk[i])=Ck[i]^2)
    Ak = A.*Wk;  %=AWk+1
    [U,S,V] = svd(Ak,'econ'); %needed to determine AWk+1 dagger = PhiG
dagger

    [d1,d2] = size(S);
    if (d1 > 1)     diag_S = diag(S);
    else            diag_S = S(1);      end;

    %NEW PART TO UPDATE LAMBDA PER ITERATION
    %find new lambda using Golden section search and Generalized Cross-
Validation
    %see section between equation (21) and (25) of "Improved M-FOCUSS
    %algorithm with overlapping blocks for locally smooth sparse signals"
    %compute GCV function with V(lambda) eq(22) is done in the (self
written) function
    %GCV.m Then the minimum lambda_hat is obtained using the Golden Section
    %Search function (fminbnd) of Matlab:
    lambda_hat = fminbnd(@(lambda_hat)
fgcv(lambda_hat,U,Y,diag_S),lambda_min,lambda_max);
    lambda_hat=real(lambda_hat); %use real values only and discard complex
values

    %now continue to calculate X using lambda_hat, according to equation
    %(18) sqrt(lambda_hat) + 1e-16
    %thikonov regularization with solving 2-norm:
PhiGdagger=G'*V*(sigma/(sigma^2+1))*U
    U_scaled = U(:,1:min(N,m)).*repmat((diag_S./(diag_S.^2 +
sqrt(lambda_hat) + 1e-16))',N,1); %the part (sigma/(sigma^2+lambda))*U =
U_scaled
    Qk_noy_withWk = Wk'.*(V*U_scaled');  %The part Wk'*V*U_scaled makes the
total Wk'*V*(sigma/(sigma^2+1))*U (=Adagger)
        %      %=Akdagger
    Xk_old = Xk; %store previous solution before writing over the new one
    Xk = Qk_noy_withWk*Y;  %=Xk+1 total solution to y=Ax using the tikhnov
SVD ==> x=Adagger*y=Xk+1 (see step 3 algortihm.)

    % *** Update hyperparameters ***
    Wk_old = Wk;    %store the old weighing matrix
    ck_nosqrt = sum(abs(Xk).^2,2); %=Ck[i] without taking the sqrt
(=Ck[i]^2)
```

```
    Wk = (ck_nosqrt/L).^(1-p/2); %=Ck[1[^1-p/2] %create the new weighing
vector: SQRT MUST STILL BE TAKEN this hapens in the first step of the new
iteration.

    % ========= Check stopping conditions, etc. =========
    count = count + 1;
    lambda_history(count,1)=lambda_hat; %store the lambdas used
    if (PRINT) disp(['iters: ',num2str(count),'   num coeffs: ',num2str(m),
...
            '   gamma change: ',num2str(max(abs(Wk - Ck_old)))]); end;
    if (count >= MAX_ITERS) break;  end;

    if (size(Xk) == size(Xk_old))
        dmu = max(max(abs(Xk_old - Xk)));
        dmu_history2(count,1)=dmu; %store the deltas between the solutions
        if (dmu < EPSILON)  break;  end;
    end;

end;


gamma_ind = sort(keep_list);
gamma_est = zeros(M,1);
gamma_est(keep_list,1) = Wk;

% expand the final solution
X = zeros(M,L);
X(keep_list,:) = Xk;

if (PRINT) fprintf('\nM-FOCUSS finished !\n'); end
return;
```

Appendix B: Effect of different random arrays.

In this thesis a pre-defined random configuration of the sparse array with 16 remaining working elements was chosen to do the simulations. When defects occur in real life this random configuration is obviously not always the same. To see if large deviations occur when other random configurations are used, simulations where done using different random arrays. This is done with 10 different random configurations. The only constraint on the random configurations is that element 1 and 64 are always working in order to keep the same aperture and hence same classical resolution for the configurations. For each random array simulations where done using 100 different on grid DOA combinations. Then for all the simulations per array the reconstructed data is compared with the original data, and the numbers of incorrect identified columns are identified. When one or more columns are incorrectly identified this is defined as a single wrong reconstruction, since not al DOAs are found. The results shown in Figure 47 are thus reconstructions in which 1 or more DOAs are wrongly estimated.

As can be seen from Figure 47 the results are quite similar for the 10 different random configurations, as long as no more than 4 targets with a maximum noise level of 20 dB arrive at the array. When the noise level is at -10dB or more than 4 targets arrive at the array (the blue dotted line and black lines) the results for different configurations deviate significantly.

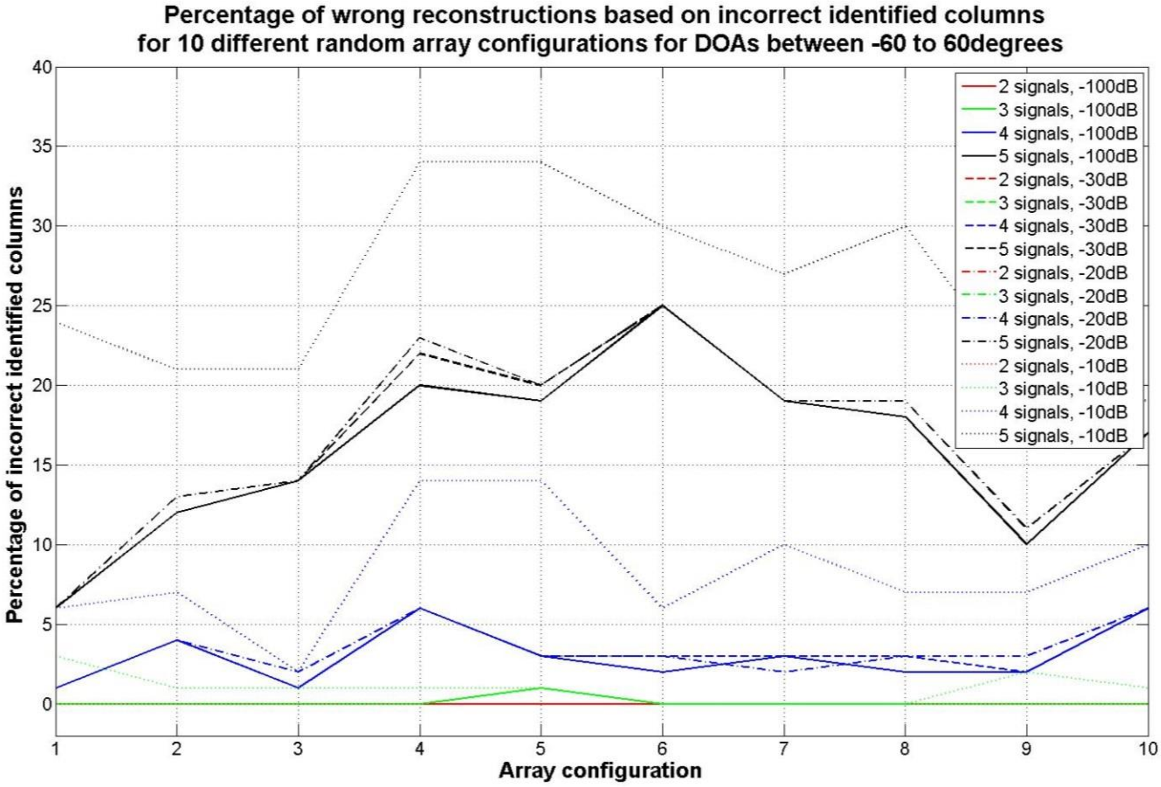

Figure 47: percentage of incorrect reconstructions based on the fact if there are incorrect identified columns for DOAs from -60 to 60 degrees.

Also the average MSE of the 100 random DOA simulations for the different array configurations is plotted in Figure 48 to see if there are large deviations. It can be seen that the MSE also starts to deviate significantly when there 4 or more targets arriving and the noise is -10 dB (the blue dotted line).

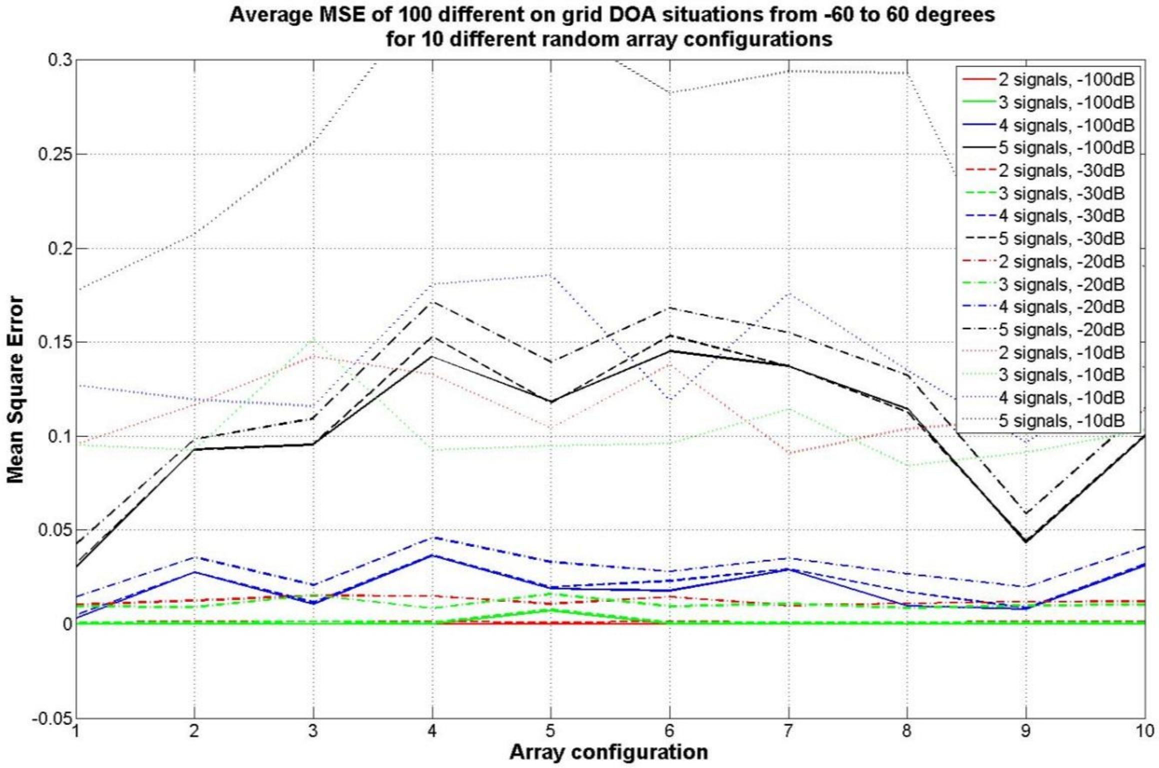

Figure 48: Average MSE over 100 different on grid DOA configurations for 10 different random arrays. The DOAs are between -60 and 60 degrees.

Although it is inevitable that different random arrays have different effects on the reconstruction results it can be said that these effects are to be discarded when no more than 4 targets arrive simultaneously with the noise level of -20dB at maximum.

Appendix C: m-file that simulates reconstruction of *K* targets.

```
%m-file to research the possibilities of using compressive sensing (with
%the MFOCUSS algorithm on a line array of 64 elements with 16 random
%working elements.
%clear the workspace and close the figures
clc; close all; clear all;
nr=0;  %simulation number to show difference when noise is different.
%       (this number sets the random generator to the specified start value)

%PARAMETER DEFINITION: CHANGE THESE TO YOUR WHISHES
%The array parameters
N=64; %Total number of elements the dense ULA
M=16; %Number of random working elements that remain in the sparse array
K=3;  %Number of targets received by the array
Nsamp = 1; %number of time snapshots/MMV used
max_scan_offbroadside=59; %define the maximum off boresight scan angle
%Define environement parameters
c = 3e8;% physconst('LightSpeed');
fc = 3e9;               % Operating frequency 3 GHz
lambda = c/fc;          % Wavelength
d=lambda/2;             % distance between elements

%Define the discretization grid
scan_angle_min=-90; %the left edge of the discretization grid
scan_angle_max=90; %the right edge of the discretization grid
coverage=2*max_scan_offbroadside;
Ns=181; %no of discretization grid points in the search area see table
below for examples:
%assumed a discretization grid from -90 to 90 degrees setting Ns to:
% 181 results in a grid spacing of 1 degree
% 361 results in a grid spacing of 0.5 degrees
% 721 results in a grid spacing of 0.25 degrees
% 1801 results in a grid spacing of 0.1 degrees
%etc

%received signal parameters
Noise=-20; %received noise level in dB
Target_power=1; % power level of the targets, choose 1 to normalize to 0dB,
% if random strength targets, this defines the maximum power level, rember
% to (un)comment line 69 for the use of random strength targets
rng(nr);
%Define the DOA of the target, give up K values between -
+max_scan_offbroadside!
DOAangles = [-50 30 35];
%or (un)comment the line below to choose K random ON GRID DOAs
%DOAangles = randperm(coverage,K)-max_scan_offbroadside %use randperm to
avoid double DOAs
% or (un)comment the 2 lines below to choose K random OFF GRID DOAs
% off_grid_distance=0.5 %specify the off grid distance for the random DOAs
% DOAangles = randperm(coverage,K)-max_scan_offbroadside+off_grid_distance
%use randperm to avoid double DOAs

%Define the MFOCUSS input parameters
lambda_m=db2pow(Noise); %regularization parameter, according to the
%description in the noisy case generally close to the noise variance. But
%also user defined values can be chosen
p=0.8; %Parameter that trades off between the speed of convergence and the
%      sparsity in the solution. p=0.8 generally leads to good results.
r=1000; %maximum number of iterations MFOCUSS can use to find a solution.
```

```
% %create a conventional ULA with 64 elements using the phased array toobox
ULA64 = phased.ULA('NumElements',N,'ElementSpacing',d);
ULA64.Element.BackBaffled = true; % this makes that the array only radiates
to one side

% For K random DOA angels:
angs = DOAangles;

%% create a signal received by the N element ULA
pos64 = getElementPosition(ULA64);
nPower = db2pow(Noise);  %noise power at the element in Watts, Power = 1; %
power of the incoming signals
sPower = Target_power; % power of the incoming signals
%sPower=Target_power*0.1*randi([1,10],1,K); %(un)comment to use random
strength signals.
% generate a multichannel signal received by the ula
rng('default');
rs = rng(0);
x64 = sensorsig(pos64/lambda,Nsamp,angs,nPower,sPower);

%% rewrite in order to use CS
%rewrite the model to X=AS+W=PSI*Z+W and solve Y=PHI*X=PHI*PSI*Z+PHI*W
X=x64.'; %this creates X; a N x T matrix.
%% Create Phi
%Create an array with only 16 working elements, but let the first and last
%one always be working setting goodrs to 11 gives the array used throughout
the
%thesis
goodrs=11;
rng(goodrs);
% first define Phi as a NxN identity matrix from which random rows can be
% removed:
Phi = eye(N); %NxN identity matrix
%% (un)comment this section this when element 1 and N may also be removed:
% rng(0);
% elemToRemove = randperm(N);  %create a row vector with random values from
1 to the total number of elements
% elemToRemove = elemToRemove(1:N-M); %Take the fist 28 values to be the
random failing elements.
% Phi(elemToRemove,:)=[]; %"randomly" make te rows of the removed elements
zero
% %take the measurements using the sensing matrix and the signal, this
% %resluts in a MxN matrix PHI which contains only values of the non broken
% %elements. using elemToRemove results in a matrix PHI which removes the
% %removed elements.
%% (Un)comment this section when element 1 and N must remain in the array
remove_range = setdiff(1:N, [1 N]); %create an array without the first and
last element number.
randomorder = randperm(length(remove_range)); % create a random number
order for elements to be removed
elemToRemove = remove_range(randomorder(1:N-M)); %remove elements at
randomorder number from the range (excl. 1&N)
elemRemain=remove_range(randomorder(N-M+1:end)); %store remaining elements
(for display purposes)
sort(elemRemain);    %sort remaining elements (exl. 1 and N)
Phi(elemToRemove,:)=[]; %remove the random rows from Phi
%% create the scanangle matrix
angles=linspace(scan_angle_min,scan_angle_max,Ns); %define the angles
representing the gridpoints
alpha(1,:)=2*pi*(d/lambda)*sind(angles); %create a row vector with the
```

```
% alpha per angle, total Ns alpha values (Ns scanangles)
%now create Psi by reproducing it for every element with the delay, Psi now
%becomes a NxNs matrix with the steering vectors for every element in a row
for n=1:N
    a_row=exp(i*alpha*(n-1));
    Psi(n,:)=a_row;  %Psi is the angle scanning matrix
end
Theta=Phi*Psi; %create the observation matrix
%create the signals received by the sparse array
Y=Phi*X; %create the MxT measurement matrix, containing time samples of M
elements
%% USE MFOCUSS to find the solution. (the file mfocuss.m needs to be in the
% same directory as this mfile, or in a "specified path" directory
[Z_MF,gamma_ind,gamma_est,iters]=mfocuss(Theta,Y,lambda_m,'p',p,'MAX_ITERS'
,r);
%% calculate the MSE with Frobenius norm (equal to the 2-norm for Nsamp=1)
max_error_MF_MSE=norm((Psi*Z_MF)-X,'fro')^2;
MSE=max_error_MF_MSE/norm(X,'fro')^2;
%% Now that CS is performed is calculated the angle spectrum Py
for theta = 1:Ns
    Py(:,theta)= 1/Nsamp * sum(abs(Z_MF(theta,:).^2));
end
%% calculate the scanned beam pattern obtained with the true X
X_meas=abs(X'*Psi);
mx=max(X_meas);
X_meas_norm=X_meas/mx;
%% Plotting: plot the scanned beam pattern and MFOCUSS solution
figure(1)
%the MFOCUSS solution:
plot(angles, pow2db(Py),'-*','MarkerSize',10,'LineWidth',1.5); grid on;
hold on;
%the (normalized) scanned beam pattern of the fully dense ULA
plot(angles,mag2db(X_meas_norm),'-r','LineWidth',1.5); hold on;
%notice to adjust the legend when one of the above lines are (un)commented
legend(['\bf\fontsize{18}MFOCUSS'],['\bf\fontsize{16}Scanned beam pattern
(X*Psi)'],'location','SE')
ylim([-40 5]); %xlim([DOAangles1(1,nnn)-10 10]);
title(['\bf\fontsize{22}',num2str(K),' targets at ',
num2str(sort(DOAangles)),' degrees. Noise level = ',num2str(Noise),'dB.']);
xlabel('\bf\fontsize{20}Broadside angle (degrees)');
ylabel('\bf\fontsize{20}Power (dB)'); axis tight; grid on;
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure.
set(gca,'FontSize',18); %set fontsize of axis numbers
%% display the location of the targets found with the very simple
% "detection algorithm" findpeaks. Note that the a priori information of
% the number of targets K is used!
[~, locs_MFOCUSS]=findpeaks(abs(Py),'NPeaks',K,'SortStr','descend');
DOA=angles(1,locs_MFOCUSS)
%display the MSE and number of iterations used by MFOCUSS
MSE
iters
```