DVB-S signal tracking techniques for mobile phased arrays.

Master's Thesis

by

Koen Blom

Committee:
prof. dr. ir. Gerard J.M. Smit
dr. ir. André B.J. Kokkeler
ir. Marcel D. van de Burgwal
ir. Kenneth C. Rovers

Computer Architecture for Embedded Systems
Faculty of EEMCS
University of Twente

December 16, 2009

Abstract

Phased array antennas are made up of multiple smaller antenna elements. Constructive interference of received waveforms results in directivity of a phased array antenna. The creation of angular regions with high sensitivity to receive signals from a desired direction is called beamforming. Mobile reception of Digital Video Broadcasting Satellite (DVB-S) signals is an interesting application of beamforming. A phased array antenna mounted on top of a vehicle should be able to electronically track the desired satellite signal during dynamic behaviour of the vehicle. This thesis discusses techniques to accomplish that goal.

The proposed system consists of a beamformer, an adaptive steerer and parts of the original DVB-S receiver. The proposed system uses phase shift based beamforming. A steering vector contains the required phase compensations to control the directivity of the phased array. The steering vector weights have to dynamically adapt to changing signal conditions. The latter is done by the adaptive steerer.

DVB-S signals contain no reference signal that can be used for steering. Therefore, the adaptive steerer uses structural properties of the signal to perform steering. Such an adaptive steerer belongs to the class of blind beamforming algorithms. The structural property of interest for a DVB-S signal is its Quadrature Phase-Shift Keying (QPSK) channel modulation. Movement of the phased array affects the beamformer output. If the phase reference of the array is centered then translational movement of the array leads to rotation of the QPSK constellation. Rotational movement of the array leads to the an orthogonal effect, a modulus decrease of the QPSK symbols. Two blind beamforming algorithms are discussed to adjust the steering vector weights based on these orthogonal effects: the Constant Modulus Algorithm (CMA) algorithm and the extended CMA algorithm. Both algorithms define a cost function that is minimized using a gradient descent.

For simulation of the proposed system vehicle dynamics are modelled to generate antenna data that contains the effects of rotational and translational movement of a vehicle. Execution of the extended CMA algorithm for this antenna data has shown the algorithm can track the desired DVB-S signal during the vehicle dynamics.

A short complexity analysis of extended CMA is performed to facilitate a later hardware implementation of the algorithm. Timing requirements are derived and it is shown that the computational complexity of extended CMA grows linear with the number of simultaneously tracked sources and the number of antenna elements.

Acknowledgements

Almost a year after I started this project I can finally finish one of the last empty sections in this report. It is quite a relieve I must say. One would expect a feeling of closure, but that is definitely not the case. In contrast, a lot of interesting ideas came up during the last weeks. I am certain that Marcel van de Burgwal and Kenneth Rovers would like to see all those ideas being realized, but sorry guys, I am running out of time now.

Throughout the whole project I got a lot of support and feedback from my committee: Gerard Smit, André Kokkeler, Marcel van de Burgwal and Kenneth Rovers. I would like to thank them for the interesting discussions, useful comments and extremely fast reviews. This thesis would never been in its present form without their help.

Of course, I would like to thank everyone from the CAES-group for creating a good atmosphere on this floor. This certainly helped me a lot during this project,

Koen Blom

Contents

1

Introduction

'Imagine yourself driving on the highway in the early morning or, being in the Netherlands, imagine yourself in a large traffic jam. Wouldn't it be nice to watch the morning news or a live stream of your company's morning meeting? However, since you got that big parabolic antenna mounted on the roof of your car gasoline seems to disappear. Not to mention the work of pointing the thing into the right direction...'

Fortunately, phased array technology can be used to deal with the problems described in the scenario above. This works starts by gradually introducing phased array antennas, but ends up with actual ideas for implementing phased array technology to enable satellite reception in vehicles.

A phased array is a type of antenna made up of multiple smaller antenna elements. Constructive interference of the received waveforms results in directivity of the array. The directivity of a phased array can be steered. Angular regions with high sensitivity can be created. Creation of those angular regions with high sensitivy to receive signals from a desired direction is called beamforming. Phased array antennas are used in radar, space communication, radio astronomy, weather research and many more applications. The use of high-speed integrated circuitry for beamforming is far from being mature and creates new opportunities for a wide range of applications [Wer08].

1.1 Research objectives

Normally Digital Video Broadcasting Satellite (DVB-S) signals are received by stationary parabolic antennas mounted to a roof or a wall. Because DVB-S signals originate from geostationary satellite sources those parabolic antennas do not have to change their pointing. However, if a parabolic antenna is mounted to a vehicle, it constantly needs re-alignment when the vehicle is moving. Phased array antennas require no satellite dish, they consist of multiple smaller antennas. Phased arrays support reception of multiple signals from different directions. The absence of a (mechanically steered) satellite dish eases opportunities for satellite reception in dynamic environments, like the reception of satellite signals in a moving vehicle. A phased array antenna mounted on top of a vehicle should be able to electronically track the desired satellite signal during dynamic behaviour of the vehicle.

Designing a system for DVB-S reception based on phased arrays requires altering of the original DVB-S receiver chain. The original DVB-S chain for

stationary parabolic antennas is shown in figure 1.1, the gray box indicates the system parts that need to be altered to support adaptive phased array technology.

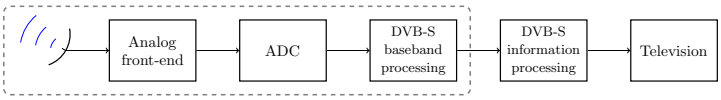

Figure 1.1: DVB-S chain for static parabolic antennas.

The desire to implement DVB-S signal tracking for mobile phased arrays leads to the main questions of this research:

- How to integrate adaptive phased array techniques in the DVB-S chain?
- Which adaptive algorithm turns out to be the most useful for coping with beam steering in dynamic (vehicle-like) situations?
- What are the requirements for an embedded platform to implement this (judged to be the) most useful adaptive algorithm?

1.2 Overview

The field of (adaptive) beamforming will be introduced by discussing fundamentals of phased array antennas in chapter 2. The phased array antenna will be mounted on top of a vehicle, so vehicle dynamics cause the actual mispointing. Vehicle dynamics are the topic of discussion in chapter 3. The adaptive beamformer should be integrated in a DVB-S receiver. Details on DVB-S and integration of an adaptive beamformer in the DVB-S chain can be found in chapter 4.

The adaptive algorithms found to be suitable for dynamic beam steering in the proposed DVB-S signal tracking system are discussed in chapter 5. Testing those adaptive algorithms for various dynamic scenarios is discussed in chapter 6. In chapter 7 a short complexity analysis is given for the adaptive algorithm that is judged to be the most useful for coping with vehicle-like dynamics. A discussion of the results is given in chapter 8. Finally, answers to the main questions can be found in chapter 9.

1.3 Notation

The following choices were made with respect to the notation of math in this thesis:

- Scalars are written in normal face lowercase letters (ex. x).
- Vectors are written bold face lowercase letters (ex. $\boldsymbol{x}$).
- Matrices are written in bold face capitals (ex. $\boldsymbol{X}$).

In block diagrams single arrows indicate real values, double arrows are used to indicate complex values and thick double arrows indicate complex vectors.

2

Beamforming

An introduction to beamforming requires basic knowledge of phased arrays. Phased array fundamentals are discussed in section 2.1. This discussion is followed by a short survey on beamforming techniques. A source can only be tracked if its initial direction is known. Therefore, Direction of Arrival (DOA) estimation is covered in section 2.3. A concise introduction to adaptive beamforming can be found in section 2.4. Conclusions and design decisions related to the subjects discussed in this chapter can be found in section 2.5.

2.1 Phased array fundamentals

The field of phased array technology is extensive, however the fundamentals discussed in this section apply to most phased array antennas. The phased array antenna described in this report is the Uniform Linear Array (ULA) type. Directivity control, calculation of the radiation patterns and mathematical modelling of a ULA are discussed in this section.

Uniform Linear Array

A ULA is made up of N adjacent antenna elements equally spaced (along a straight line) at a distance d apart. Antenna elements are capable of transmitting and receiving electromagnetic waveforms. The elements are assumed to be isotropic, which means that their reponse is uniform regardless of the signal direction.

A ULA exhibits certain directivity properties. Directivity is a preference for certain directions over others. This phenomenon is the result of the interference of electromagnetic waveforms. Interference effects cause a maximal sensitivity for waveforms perpendicular to the array of antenna elements. Signals perpendicular to the phased array axis are in-phase, so there will be no destructive interference between the waveforms. Interference effects can be used to steer the directivity of the antenna array.

Directivity control

Consider the antenna array from figure 2.1. A signal is arriving at the array with angle θ from the normal of the array. The received wavefront is assumed to be planar.
Assume that the leftmost element is the reference element. The waveform received at the second leftmost element has a difference in path length of $d \cdot \sin(\theta)$.

When the waveform satisfies the narrowband assumption this difference in path length leads to a phase shift of the waveform equal to $2\pi(d \cdot \sin(\theta)/\lambda)$. Herein, λ is the wavelength of the received signal. The term narrowband is used for signals whose bandwidth is much smaller than their center frequency [PM96]. The Fractional Bandwidth (FB) is used to verify if a signal satisfies the narrowband assumption [AG05]. The FB can be written as:

$$FB = \frac{f_h - f_l}{(f_h + f_l)/2} \cdot 100\% \tag{2.1}$$

Herein, f_h and f_l are the highest and the lowest frequency components respectively. A signal satisfies the narrowband assumption if the FB is less than 1% [AG05].

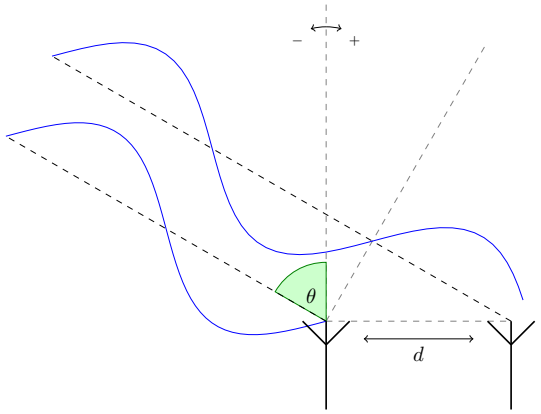

Figure 2.1: Phase delay

The main advantage of a narrowband signal is that it can be treated as a sinusoidal signal. The narrowband electromagnetic signal with wavelength λ that arrives at the n^{th} antenna element has a phase shift ϕ_n of:

$$\phi_n = 2\pi \left(\frac{d \cdot \sin(\theta)}{\lambda} \right) \cdot n \quad , n \in [0 \ldots (N-1)] \tag{2.2}$$

The directivity of the linear array can be steered by introducing a phase shift $-\phi_n$ for each element to compensate for the phase shift ϕ_n. The combination of all the shifted signals can be done coherently, resulting in a 'beam' in direction θ [Sko01].

Various other beamforming techniques can be found in section 2.2. Beamforming techniques also exist for non-uniform and two-dimensional arrays [AG05], but those are not discussed in this report.

Radiation patterns

The sensitivity of the combined antenna signals has a directional dependence even if all the separate antenna elements are isotropic. The normalised amplitude of the signal after beamforming plotted against all possible angular positions

is called the radiation pattern. The total radiation pattern of a linear array depends on the pattern of each individual element (element factor) and the pattern of an array of elements (array factor). The complex signal received by the n^{th} antenna element of the linear array can be written as [Vis05]:

$$S_n(\theta) = S_{e_n}(\theta) a_n e^{j\phi_n} \tag{2.3}$$

Herein, a_n is the gain of the n^{th} element. The term $S_{e_n}(\theta)$ represents the directional sensitivity of the n^{th} element. The phase shift ϕ_n for each element is expressed by the complex exponential $e^{j\phi_n}$ and can be determined using equation 2.2. Usually the individual elements have low directivity, therefore the total radiation pattern depends mostly on the array factor [Rud82]. For now assume the gains a_n and the element factors $S_{e_n}(\theta)$ to be equal to one. The array factor can then be written as:

$$S_a(\theta) = \sum_{n=0}^{N-1} S_{e_n}(\theta) a_n e^{j\phi_n} = \sum_{n=0}^{N-1} e^{j\phi_n} \tag{2.4}$$

The normalised (logarithmic) power pattern can be found by normalizing for all N antenna elements:

$$P_a(\theta) = 20 \log_{10}(|S_a(\theta)| / N) \tag{2.5}$$

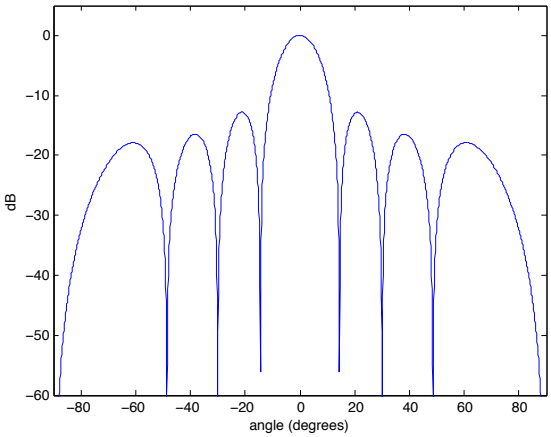

Figure 2.2: Eight element ULA power pattern with $d = \frac{1}{2}\lambda$.

The normalized power pattern for an eight element array with equally spaced elements at a distance $\frac{1}{2}\lambda$ apart can be seen in figure 2.2.

A lobe is an angular region of strong radiation [Vis05]. The lobe in the direction of the highest sensitivity is called the main lobe. Other smaller radiation lobes are called sidelobes. Due to interference of the impinging waveforms the array is not sensitive in certain directions. Those directions are called nulls.

Reduction of the spacing between antenna elements leads to a smaller number of lobes, but each individual lobe will be wider. A spacing $d > \frac{1}{2}\lambda$ leads

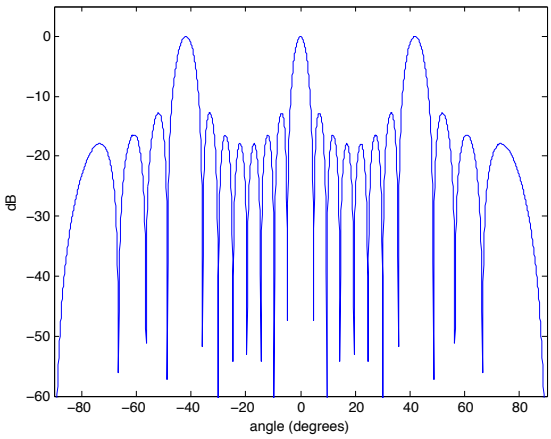

Figure 2.3: Eight element ULA power pattern with $d = 1\frac{1}{2}\lambda$.

to an often undesirable situation with additional sidelobes that have the same sensitivity as the main beam, these lobes are called grating lobes. In figure 2.3 the power pattern of an eight element ULA is shown for $d = 1\frac{1}{2}\lambda$. In this particular situation two grating lobes appear in the power pattern.

The shape of the main lobe is often indicated using the half-power beamwidth [Vis05]. This is the angular separation on the main lobe where the received power is half that of the maximum received power. It can be found by looking at the points where the power transfer changes $10\log_{10}(0.5) = -3$ dB from the maximum received power. The half-power beamwidth of an eight element ULA with $d = \frac{1}{2}\lambda$ is shown in figure 2.4.

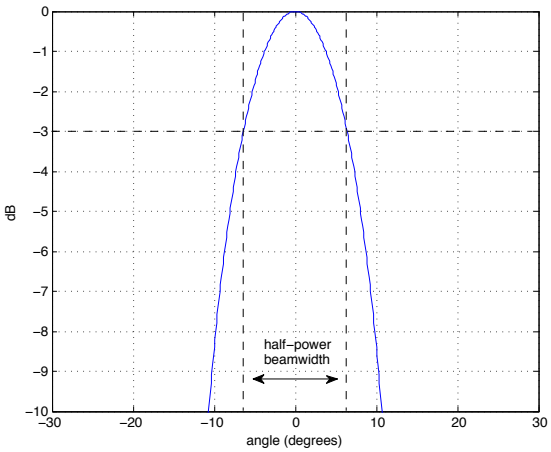

Figure 2.4: Half-power beamwidth.

Data model

The wavefronts impinging at the linear array can be represented in a data model. An understanding of (adaptive) phased array processing algorithms requires

knowledge of this data model. An intuitive understanding of the data model is given based on figure 2.5, which shows a situation with two sources signalling a three antenna element array. Assume enough distance between the sources and the elements to consider the arriving waveforms to be planar.

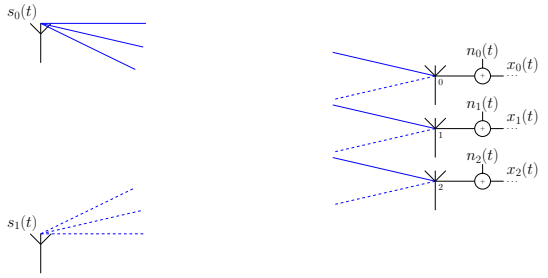

Figure 2.5: Signals impinging at a ULA.

The source signals are represented by a vector $\boldsymbol{s}(t)$. For every antenna element the received signal depends on the angle with respect to the source and on the position of the antenna element in the linear array. The effects of those source angles and array positions are captured in a matrix denoted by $\boldsymbol{A}$ in equation 2.6.

Until now the effects of noise have been disregarded. Noise influences the correctness of the DOA estimation algorithm. Therefore signal and instrumentation noise $\boldsymbol{n}(t)$ are added to the data model. The antenna signals $\boldsymbol{x}(t)$ after addition of noise can be written as [Sch86]:

$$\begin{aligned}\boldsymbol{x}(t) &= \boldsymbol{A}\boldsymbol{s}(t) + \boldsymbol{n}(t) \\ &= \begin{pmatrix} a_{0,0} & \cdots & a_{0,(k-1)} \\ \vdots & & \\ a_{(N-1),0} & \cdots & a_{(N-1),(k-1)} \end{pmatrix} \begin{pmatrix} s_0(t) \\ s_1(t) \\ \vdots \\ s_{(k-1)}(t) \end{pmatrix} + \begin{pmatrix} n_0(t) \\ n_1(t) \\ \vdots \\ n_{(N-1)}(t) \end{pmatrix}\end{aligned} \tag{2.6}$$

Herein, k is the number of signal sources and N is equal to the number of antenna elements. A large number of satellite broadcasting models use Additive White Gaussian Noise (AWGN) to model noise [Gom02]. For that reason the noise added to the received signal is white with a Gaussian amplitude distribution.

2.2 Beamforming techniques

Three common methods to perform beamforming are discussed in this section. Phase shift based beamforming is used in the suggested adaptive beamformer and is therefore covered in more detail.

Time delay

Introducing time delays in each of the array elements can be used to steer the beam in a particular direction. The time delays have to compensate for the

real time delay experienced by the array elements when the waveform arrives. Mathematically the time delay τ_n experienced by the n^{th} antenna element of a ULA is written as:

$$\tau_n = \left(\frac{d \cdot \sin(\theta)}{p} \right) \cdot n \tag{2.7}$$

Herein, p is the propagation speed of the waveform. For radio transmissions p can be considered equal to the speed of light. The first array element that receives the waveform from direction θ should compensate for the time that it takes for the last array element to receive the waveform from that specific direction. The time compensation can be introduced using a delay line or a buffer. Summation of all time delayed signals results in the signal $y(t)$ from one specific direction θ:

$$y(t) = \sum_{n=0}^{N-1} x_n(t - \tau_n) \tag{2.8}$$

The advantage of time delay based beamforming is that delayed signals add in-phase for every frequency component of the original signal. Therefore, this technique is useful in wideband applications.

Phase shift

Beamforming based on phase shifting can only be performed when the narrowband assumption is guaranteed. Phase shifting a narrowband signal acts like a shift of the signal in time. The error of this time shift increases when the frequency of the signal to be shifted is further away from its center frequency. Phase information should be reconstructed from the original real signal before beamforming based on phase shifting can be performed. Phase information reconstruction of the real antenna signal can be accomplished by a Hilbert transform, this is explained in the following section. The actual phase shift can be implemented by complex multiplication.

Hilbert transform

The Hilbert transform creates a complex signal from the original signal by discarding its negative frequency components. A real sinusoid consists of an equal contribution of positive and negative frequency components:

$$\cos(\omega t) = \frac{1}{2}(e^{j\omega t} + e^{-j\omega t}) \tag{2.9}$$

The complex sinusoid $Ae^{j\omega t}$ makes many mathematical manipulations of the signal easier to perform because it only contains the positive frequency ω. By looking at Euler's formula $Ae^{j\omega t} = A(\cos(\omega t) + j\sin(\omega t))$ it can be seen that an in-phase and phase-quadrature component are necessary to represent the complex sinusoid $Ae^{j\omega t}$. Thus, creation of the complex sinusoid requires generation of the phase-quadrature component, the imaginary part of the signal. The complex representation of a real signal can be found using the Hilbert transform [Ste00]. The Hilbert transform performs a $-\frac{1}{2}\pi$ phase shift on the

positive frequency components and a $\frac{1}{2}\pi$ phase shift on the negative frequency components, which can be mathematically written as follows:

$$H(\omega) = \begin{cases} -j & \omega > 0 \\ 0 & \omega = 0 \\ j & \omega < 0 \end{cases} \tag{2.10}$$

As an example [Smi07], the Hilbert transform is applied to the real signal $x(t)$. The result of the transform is the quadrature signal $y(t)$. Both $x(t)$ and $y(t)$ are used the describe the complex signal $z(t)$.

$$\left.\begin{array}{l} x(t) = e^{j\omega t} + e^{-j\omega t} \\ y(t) = e^{-j\pi/2}e^{j\omega t} + e^{j\pi/2}e^{-j\omega t} = -je^{j\omega t} + je^{-j\omega t} \\ z(t) = x(t) + jy(t) \end{array}\right\} \Rightarrow z(t) = 2e^{j\omega t} \tag{2.11}$$

Equation 2.11 shows that the negative frequency component of the real signal disappears and the amplitude of the positive frequency component doubles. The latter can be explained by the desire to retain the original energy after removal of the negative spectral components [Ste00].

The Hilbert transform can be implemented as a Finite Impulse Response (FIR) filter. To determine the filter coefficients the inverse Fourier transform of the Hilbert frequency response is taken. A comprehensive derivation of this impulse response is stated in [Lyo99]. Assume f_s to be the sampling frequency, then the discrete Hilbert impulse response can be written as:

$$h(n) = \begin{cases} \frac{f_s}{\pi n}(1 - \cos(\pi n)) & n \neq 0 \\ 0 & n = 0 \end{cases} \tag{2.12}$$

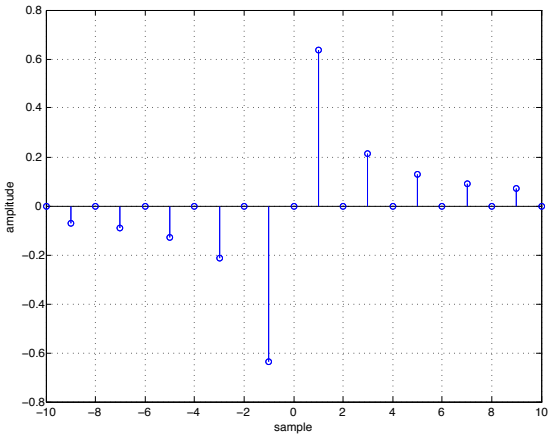

Figure 2.6: Discrete impulse response of the Hilbert transform.

The Hilbert filter needs samples from the past to calculate current quadrature signal values, therefore a delay is introduced to the in-phase signal to stay synchronized with the quadrature signal. This delay is called a group delay and its size depends on the number of taps of the FIR filter.

In [Lyo99] is shown that an odd-tap FIR implementation of the Hilbert transform provides some advantages compared to an even-tap filter. A look

at figure 2.6 of the Hilbert impulse response shows zero transfers for the even time steps. These can be used to reduce the number of multiplications in the odd-tap filter without loss of accuracy.

Complex multiplication

The actual phase shift of the antenna signals is performed by complex multiplication of the complex signals by a steering vector $\boldsymbol{\phi}$ consisting of compensations for the phaseshifts shown in equation 2.2. The magnitude of the complex numbers in the steering vector is equal to one. Thus the original signal amplitude does not change. Recall that the leftmost antenna element is assigned index 0 and ϕ_n is the phase shift for the antenna element with index n. The first element of the steering vector is equal to one, since that particular signal does not need a phase shift. The steering vector $\boldsymbol{\phi}$ can be written as:

$$\boldsymbol{\phi}(\theta) = \begin{pmatrix} 1 \\ e^{-j\phi_1} \\ e^{-j\phi_2} \\ \dots \\ e^{-j\phi_{n-1}} \end{pmatrix} \tag{2.13}$$

In literature phase shift based beamforming is written in two different forms, the used form depends on the authors definition of the steering vector $\boldsymbol{\phi}$. If the steering vector $\boldsymbol{\phi}$ contains phase shift compensations (as in equation 2.13) then beamforming can be written as [TA83]:

$$y = \boldsymbol{\phi}^T \boldsymbol{x} \tag{2.14}$$

Herein, $\boldsymbol{x}$ is the complex antenna snapshot and y the result after beamforming. If $\boldsymbol{\phi}$ contains the phase shifts that occur for each antenna element then beamforming is written as [vdV04]:

$$y = \boldsymbol{\phi}^H \boldsymbol{x} \tag{2.15}$$

Equation 2.15 calculates the beamformer output y by multiplication of the complex antenna snapshot $\boldsymbol{x}$ by the Hermitian of the steering vector $\boldsymbol{\phi}$. The Hermitian is the complex conjugate transpose of a vector. The complex conjugate leads to the required phase compensations.

The steering vector $\boldsymbol{\phi}$ needs to be converted from polar to Cartesian form before its values can be multiplied with the complex antenna signals. Euler's formula can be used to perform this conversion ($Ae^{j\omega t} = A(\cos(\omega t) + j\sin(\omega t))$). Multiple methods exist to calculate trigonometric functions, this is discussed in section 7.1. If both signals are in Cartesian form then complex multiplication is written as:

$$(a + bj)(c + dj) = (ac - bd) + (ad + bc)j \tag{2.16}$$

Complex multiplication requires two additions and four multiplications if it is implemented as shown in equation 2.16. Several methods exist to implement complex multiplication. An overview of different implementations can be found in [Lyo04].

Fast Fourier Transform

Assume a ULA that consists of N antenna elements. For each sample moment all the N samples of the elements are passed to an N-point complex Fast Fourier Transform (FFT) at once. Because of the simultaneous sampling of the equally spaced elements the FFT can now be considered as a 'spatial' FFT. The input samples are separated in space (instead of time), which results in output samples that are separated in direction (rather than in frequency) [Hay98]. For example, assume a linear array that consists of three elements where the planar waves arrive perpendicular to the array, as shown in Figure 2.7.

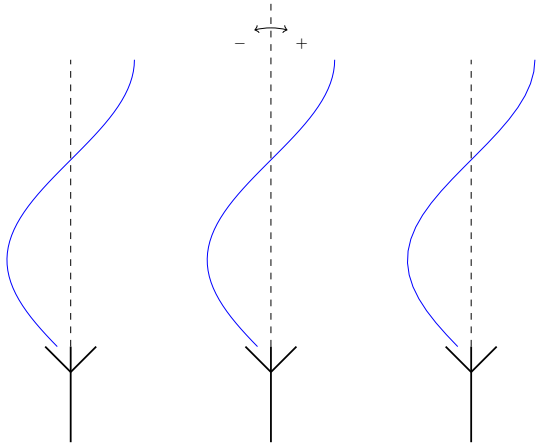

Figure 2.7: Waves arriving perpendicular to the array.

A perdendicular waveform leads to equal sample values for all antenna elements when the sampling occurs simultaneously. The FFT of those samples will only contain a zero frequency component. Now assume that the direction of arrival (DOA) changes to an angle θ of -60 degrees. This can be seen in Figure 2.8. In this case each radiator produces a different value when sampled simultaneously. In fact a sinusoidal pattern in the spatial domain is generated.

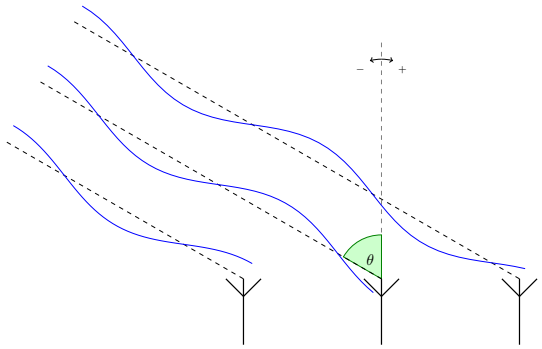

Figure 2.8: Waves arriving at an angle θ of -60 degrees.

An increase of the absolute value of the angle θ results in higher frequencies of the sinusoidal pattern generated by spatial sampling. Certain ranges of equally

spaced arrival angles are mapped to equally spaced frequency bands. An FFT produces as many outputs as the number of inputs. Thus, the number of beams depends on the number of antenna elements. There is a fixed position relation between beams, if one beam is steered in a certain direction then the positions of other beams are restricted. Pre-steering can be applied to compensate for this restriction and steer the beam in the exact direction. Pre-steering is complex multiplication with pre-calculated weight factors to shift the angular region of the array pattern with the highest sensitivity exactly in the direction of the source.

The FFT based beamforming technique can be implemented in both digital or analog hardware. A description of its analog implementation can be found in [Sko01]. This beamforming technique is also called Butler beamforming.

2.3 Direction of Arrival estimation

A DOA estimation algorithm calculates the initial steering angle to be used by the tracking algorithm. The following section discusses Multiple Signal Classification (MUSIC) [Sch86], a high resolution DOA estimation algorithm.

MUSIC

The MUSIC algorithm can be used to provide DOA estimates of the incoming signals based on complex antenna snapshots.

MUSIC uses eigenvalues and eigenvectors of the input signal autocorrelation matrix $\boldsymbol{S}$ to calculate the arrival angles of incoming signals. First, the autocorrelation matrix $\boldsymbol{S}$ is calculated. It describes the degree of correlation that exists between elements of two equal snapshot vectors. Every radiator value is compared to all other radiator values. The autocorrelation matrix for m complex snapshot vectors is described by [Sch86]. Note that every complex snapshot vector $\boldsymbol{x}[k]$ contains N values.

$$\boldsymbol{S} = \frac{1}{m} \sum_{k=1}^{m} \boldsymbol{x}[k] \boldsymbol{x}^H[k] \tag{2.17}$$

In literature the term autocorrelation is interchangeably used with autocovariance. Furthermore, autocorrelation $\boldsymbol{R}_{xx}$ is a special form of cross-correlation where the cross-correlated vectors are the same. An in-depth discussion of those concepts can be found in [MM04]. The term $\boldsymbol{x}^H[k]$ denotes the Hermitian of $\boldsymbol{x}[k]$. The Hermitian is calculated by taking the transpose of $\boldsymbol{x}[k]$ after complex conjugation.

The autocorrelation matrix $\boldsymbol{S}$ ($N \times N$ elements) is Hermitian, thus the complex conjugate transpose of $\boldsymbol{S}$ is equal to $\boldsymbol{S}$ itself. For an Hermitian matrix it is possible to find an orthonormal basis that consists of only eigenvectors. Once this basis is found the l largest eigenvectors (and their corresponding eigenvalues) are considered to denote the sources. The $(N - l)$ smallest eigenvectors are assigned to the noise. The white noise sources should be treated to be statistically independent of the signal sources.

A linear combination of the largest eigenvectors describes the signal subspace. Symmetrically, a linear combination of the smallest eigenvectors of the matrix

$\boldsymbol{S}$ describes the noise subspace $\boldsymbol{E}_N$. Orthogonal vectors can be treated to be statistically independent, therefore the vectors as orthogonal as possible to the noise subspace indicate the source directions. Orthogonality can be checked by looking at the inner product of the noise subspace $\boldsymbol{E}_N$ and the array manifold $\boldsymbol{\rho}(\theta)$. Whenever the inner product of the noise subspace and the array manifold $\boldsymbol{\rho}(\theta)$ goes further to zero the further the signal raises above the noise floor. The complex inner product of $\boldsymbol{\rho}(\theta)$ and $\boldsymbol{E}_N$ can be found as followed:

$$\langle \boldsymbol{\rho}(\theta), \boldsymbol{E}_N \rangle = \boldsymbol{\rho}(\theta)^H \boldsymbol{E}_N \tag{2.18}$$

Given the noise subspace $\boldsymbol{E}_N$ and the complex array manifold $\boldsymbol{\rho}(\theta)$ the MUSIC spectrum P_{MU} can be drawn. The squared length of (the modulus of) the inner product is used in the denominator of the MUSIC spectrum formula [God04], that can be written as follows:

$$P_{MU}(\theta) = \frac{1}{\left|\boldsymbol{\rho}_\theta^H \boldsymbol{E}_N\right|^2} \tag{2.19}$$

The smaller the inner product in the denominator of Equation 2.19 becomes, the higher the peaks in the MUSIC spectrum are. Multiple peaks denote multiple sources. A mathematical analysis of MUSIC can be found in [Sch86].

2.4 Adaptive beamforming algorithms

The dynamic behavior of a vehicle influences the arrival angle of the impinging satellite signal. The array steering vector $\boldsymbol{\phi}$ that controls the direction of the beam has to adapt its values to keep track of the signal whenever the vehicle experiences movement. Beamforming algorithms that adapt steering vector weights due to changing signal conditions are called adaptive beamforming algorithms. Various of those adaptive algorithms exist to find new values for the steering vector in dynamic situations [AG05]. The class of adaptive array algorithms consists of three subclasses [AG05]:

- Temporal reference beamforming algorithms
- Spatial reference beamforming algorithms
- Blind beamforming algorithms

Temporal reference beamforming algorithms use cross-correlation with known temporal signal properties to adjust the weights of the array steering vector $\boldsymbol{\phi}$. An in-depth discussion on temporal reference beamforming can be found in [AG05].

The use of DOA estimation algorithms to detect angles of impinging signals based on antenna snapshots is called spatial reference beamforming. An antenna snapshot is the result of simultaneous sampling of all the N antenna elements. The desired signal is chosen out of the set of all angles found by the DOA algorithm, temporal techniques may be used to improve this decision. The MUSIC algorithm mentioned in section 2.3, is an example of a spatial reference algorithm. Most spatial reference algorithms are inappropriate for real-time weight adjustment because of their high computational costs.

The class of blind beamforming algorithms uses known structural and statistical properties of the desired signal to determine the arrival angle of the signal. DVB-S signals are Quadrature Phase-Shift Keying (QPSK) modulated. The structural properties of a QPSK modulated signal are exploited for blind beamforming.

2.5 Conclusion

Various beamforming methods are mentioned in this chapter. The beamforming technique for the proposed adaptive beamformer is phase shift based beamforming. This particular technique is chosen because time delay based beamforming is hard to implement in digital hardware and FFT based beamforming has fixed position relations between beams. Furthermore, complex multiplication (which is often used) to implement phase shift based beamforming is a common operation in most embedded platforms.

The class of blind beamforming algorithms seems the most appropriate for adaptive beamforming of DVB-S signals due to the constant modulus property of DVB-S signals. Temporal techniques cannot be used for DVB-S signals, because there is no reference signal in these signals. DOA estimation algorithms are inappropriate because of their high computational costs. Properties of DVB-S signals are discussed in chapter 4. A discussion of two algorithms that exploit DVB-S properties can be found in chapter 5.

3

Modelling vehicle dynamics

The dynamics of a vehicle can be expressed by mathematical models. Two different vehicle models are discussed in this chapter. The models are used to calculate DOA angle dynamics to test the performance of the adaptive array algorithms discussed in chapter 5.

3.1 Degrees of freedom

A moving vehicle experiences both translational and rotational motion. It is assumed that the wavefront is planar, therefore only rotational motion influences the DOA of the received signal. A vehicle experiences rotational movement in three degrees of freedom. The positive directions of rotational movement and their appropriate names are mentioned in figure 3.1.

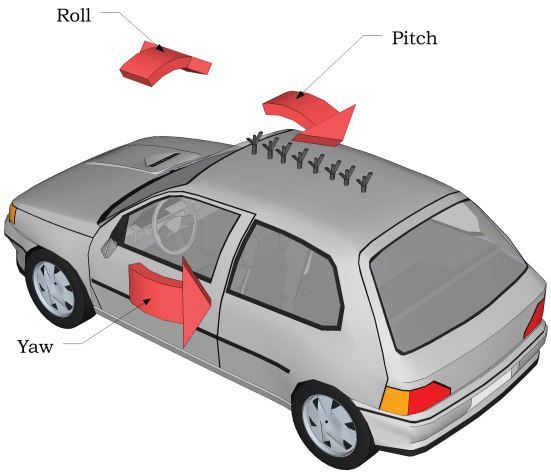

Figure 3.1: ULA parallel to the direction of driving [G3D09].

Only pitch and yaw motion lead to DOA changes of the received signal. If all antenna elements have an isotropic element factor then rolling motion does not affect the sensitivity of the received signal.

3.2 Vehicle models

The following sections discuss models to describe yawing and pitching of vehicles. Yawing motion is modelled based on the planar bicycle model. The simulations of pitching motion are based on a half-car suspension model.

Planar bicycle model

A tire generates lateral forces as it deforms when its direction of travel is different from its longitudinal axis. The angle between the longitudinal axis x_t of the front tire and the velocity vector $\boldsymbol{v}_f$ is called the sideslip angle α_f. Furthermore the angle between the x axis of the body (indicated by the dotted line) and the velocity vector of the front tire $\boldsymbol{v}_f$ is called indicated by the angle β_f. The tire is steered by steering angle δ. The situation is sketched in figure 3.2 [Jaz08].

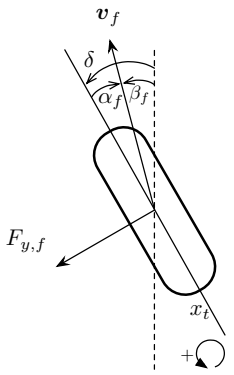

Figure 3.2: Generation of a lateral force due to deformation.

Figure 3.2 can be interpreted as followed, a positive steering angle δ of a tire moving in a forward direction generates a negative sideslip angle α_f. The relation between the lateral force supplied by the front tire as function of small slip angles is given by [Jaz08]:

$$F_{y,f} = -C_{\alpha,f}(\beta_f - \delta) = -C_{\alpha,f}\alpha_f \tag{3.1}$$

$C_{\alpha,f}$ represents the cornering stiffness of the front tire, which can be determined by experimental measurements. The rear tire of a car cannot change the angle of its longitudinal axis, therefore the slip angle of the rear tire only depends on the direction of its velocity vector $\boldsymbol{v}_r$.

In typical driving situations the lateral forces for the inside and outside wheels are approximately the same. Therefore the planar forward, lateral and yawing motion of a vehicle can be modeled using the bicycle model. The latter is a much used representation in analyzing vehicle dynamics. The planar bicycle model can be seen in figure 3.3. The x-axis is a longitunal axis directing from the back to the front of the vehicle. The y-axis is directed from the right to the left from the drivers viewpoint. The z-axis points upward from the centre of gravity of the vehicle, opposite to the gravitational acceleration of earth [Jaz08].

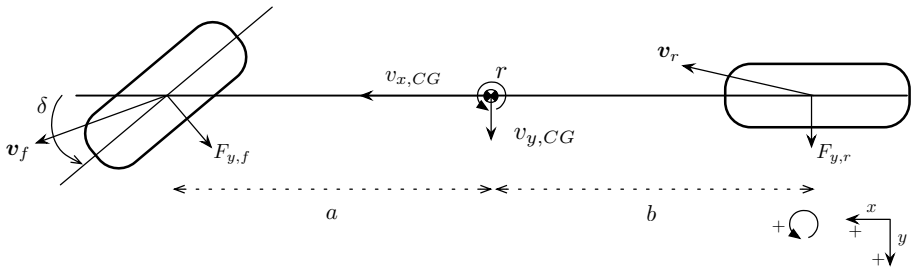

Figure 3.3: Bicycle model.

Small slip angles α_f of the front tire can be expressed in terms of the longitudinal velocity, lateral velocity and the steering angle δ of that tire. An analogue derivation is valid for the rear tire, except that its slip angle α_r is not influenced by the steering angle δ [Jaz08]:

$$\begin{aligned} \alpha_f &= \beta_f - \delta = \tan^{-1}\left(\frac{v_{y,CG}+a\cdot r}{v_{x,CG}}\right) - \delta \approx \left(\frac{v_{y,CG}+a\cdot r}{v_{x,CG}}\right) - \delta \\ \alpha_r &= \tan^{-1}\left(\frac{v_{y,CG}-b\cdot r}{v_{x,CG}}\right) \approx \left(\frac{v_{y,CG}-b\cdot r}{v_{x,CG}}\right) \end{aligned} \tag{3.2}$$

Analysing the dynamic behaviour of a vehicle during cornering requires a set of differential equations based on the force and moment balance of the bicycle model. An in-depth discussion of deriving this set of differential equations is not relevant for this research. An extensive derivation of the equations can be found in [Jaz08]. The derivation is based on the assumption that the vehicle does not accelerate during cornering ($\dot{v}_{x,CG} = \frac{dv_{x,CG}}{dt} = 0$) and is only valid for small slip angles, because an approximation for the arctangent is used in equation 3.2. The result of the derivation is the following set of state equations that describes the dynamic behaviour of the yaw rate r and the lateral velocity $v_{y,CG}$ [Jaz08]:

$$\begin{vmatrix} \dot{v}_{y,CG} \\ \dot{r} \end{vmatrix} = \begin{vmatrix} -\frac{C_{\alpha,f}+C_{\alpha,r}}{M\cdot v_{x,CG}} & \frac{-aC_{\alpha,f}+bC_{\alpha,r}}{M\cdot v_{x,CG}} - v_{x,CG} \\ -\frac{aC_{\alpha,f}-bC_{\alpha,r}}{I_y\cdot v_{x,CG}} & -\frac{a^2C_{\alpha,f}+b^2C_{\alpha,r}}{I_y\cdot v_{x,CG}} \end{vmatrix} \begin{vmatrix} v_{y,CG} \\ r \end{vmatrix} + \begin{vmatrix} \frac{C_{\alpha,f}}{M} \\ \frac{aC_{\alpha,f}}{I_y} \end{vmatrix} \delta \tag{3.3}$$

Suspension model

The pitching behavior of a vehicle can be modeled using a half-car suspension schematic. The schematic is useful for calculation of the (pitch) natural frequencies of the suspension [RSF01]. Natural frequencies of cars are objective measures for ride quality.

The pitch natural frequency affects the reception of satellite signals by the ULA, that is mounted on the roof of the vehicle. Several variables affect the pitch natural frequency. It is a function of the pitch moment of inertia, the position of the center of gravity and the suspension and tire stiffnesses. Further analysis is based on the half-car suspension model that can be seen in figure 3.4 [RSF01]. Herein, K_{tf} and K_{tr} represent the front and rear tire stiffnesses.

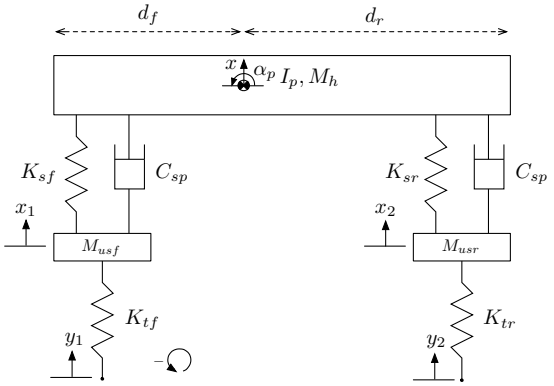

Figure 3.4: Half-car suspension schematic.

Time dependent displacements y_1 and y_2 are applied to the tires while driving. The mass of the suspension and the wheels themselves are represented by M_{usf} and M_{usr}. M_h indicates half the mass of the car body. The rear and front suspension both consist of a spring and damper connected to the car body. The inertia of half the car body is drawn as I_p and its location given by d_f and d_r. α_p indicates the pitch angle of the car body. Practical suspension damping ratios range from 0.2 to 0.3. Pitch natural frequencies should stay below 1-2 Hz, because those values already correlate with passenger discomfort. In [Jaz08] a state space description of figure 3.4 is derived that can be used to simulate pitching behaviour, this state space description can also be found in appendix A.

3.3 Analysis of driving scenarios

Chapter 6 discusses the use of dynamic models for yawing and pitching to analyse convergence behavior of adaptive steering algorithms. Simulation of the algorithms requires realistic parameters for the dynamic models, the parameters used in this work are from a Renault Clio RL 1.1 [SGF07]:

Vehicle mass (kg)	825
Passenger mass (kg)	75
Distance from center of gravity to front axes (m)	0.916
Distance from center of mass to rear axes (m)	1.556
Yaw inertia ($kg\ m^2$)	2345
Sideslip coefficient of front tires ($Nrad^{-1}$)	$60 \cdot 10^3$
Sideslip coefficient of rear tires ($Nrad^{-1}$)	$60 \cdot 10^3$
Pitch inertia ($kg\ m^2$)	2443
Undamped mass (front and rear) (kg)	38.42
Tire stiffness (front and rear) (N/m)	$150 \cdot 10^3$
Suspension stiffness (front and rear)(N/m)	$14.9 \cdot 10^3$
Suspension damping coefficient (front and rear) (Ns/m)	475

Table 3.1: Parameters of the Renault Clio RL 1.1 [SGF07].

Sudden change in steering angle

The first simulation scenario can be found in [Jaz08]. The dynamics of the Renault are considered during a instantaneous steering angle of 11.5 degrees while driving at a forward velocity of 72 km/h. The Renault Clio with a ULA mounted on the roof (parallel to the cars longitudinal axis) can be seen in figure 3.5. Symbol ψ is used to indicate the heading angle of the car. The DOA of the signal wavefront is indicated by θ. Positive orientations of ψ and θ are shown in the leftmost drawing of figure 3.5. The angle ψ is equal (in value) to the DOA of the incoming signal.

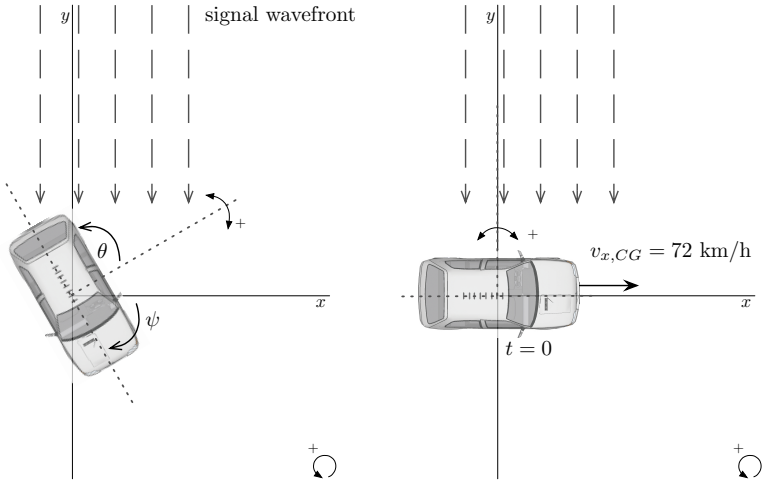

Figure 3.5: Angular references and positive orientations for modeling yawing.

Simulation is started at a heading angle ψ_0 of zero degrees and a forward velocity of $v_{x,CG}$ equal to 72 km/h. The initial situation is shown in the rightmost drawing of figure 3.5.

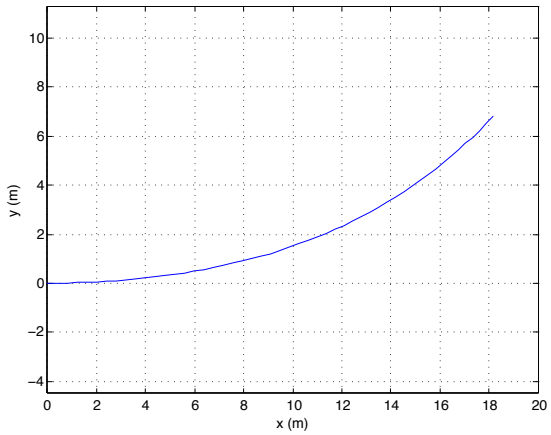

Figure 3.6: Course of the car during the steering manoeuvre.

Mapping the motion of the car (body fixed frame) expressed by r, $v_{x,CG}$ and

$v_{y,CG}$ to its position in the global coordinate frame (shown in figure 3.5) can be done by applying the following coordinate transformation [Jaz08]:

$$\begin{aligned} \psi &= \psi_0 + \int r dt \\ x &= \int (v_{x,CG}\cos(\psi) - v_{y,CG}\sin(\psi))dt \\ y &= \int (v_{x,CG}\sin(\psi) + v_{y,CG}\cos(\psi))dt \end{aligned} \tag{3.4}$$

From a global coordinate frame perspective a sudden change in steering angle leads to a course change. The path followed by the car during the first second of the steering manoeuvre is shown in figure 3.6.

Changes to the car's yaw rate r during the steering manoeuvre are shown in the leftmost graph of figure 3.7. A typical second-order step response can be recognized. The amount of overshoot depends on the longitudinal velocity $v_{x,CG}$ of the car in the initial situation. The constant yaw rate that eventually will be reached depends on the size of the sudden steering angle.

After integration of the yaw rate the new heading angle is found ($\psi(t) = \psi_0 + \int r dt$). The start up effect of the sudden steering angle manoeuvre can be recognized in the deviation of the heading angle from its linear approximation.

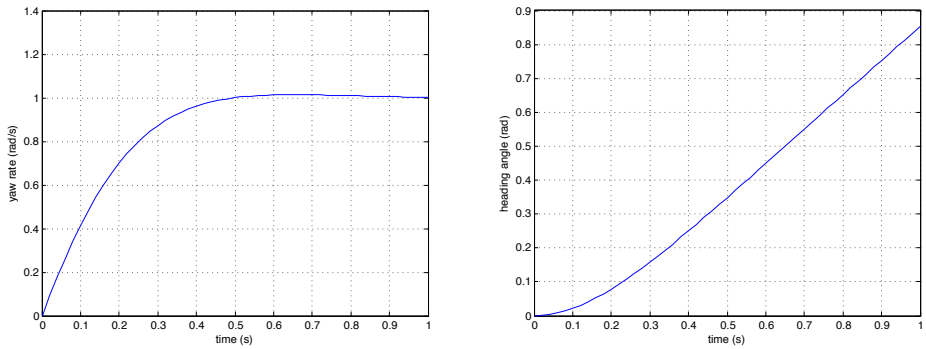

Figure 3.7: The yaw rate r and heading angle ψ during the manoeuvre.

A DVB-S satellite has a fixed position in the sky. Thus, yawing movement of the car should be compensated by the adaptive array. In figure 3.5 it can be seen that the angle ψ is equal (in value) to the DOA of the incoming signal. Therefore, the heading angle from figure 3.7 is taken as a reference DOA in simulations of chapter 6.

The velocity of the car towards the source satellite causes Doppler effects, which are explained in chapter 4. Figure 3.8 shows the car's velocity v_y (in the global coordinate frame) towards the source satellite for the sudden steering angle scenario.

Sudden road excitation

Changes in the vehicle body pitch angle α_p influences the DOA of a satellite signal received by the phased array mounted longitudinally on the roof of the vehicle. The half-car suspension model is used to model vehicle pitching. The half-car model was already discussed in section 3.2 and equations of this model can be found in appendix A.

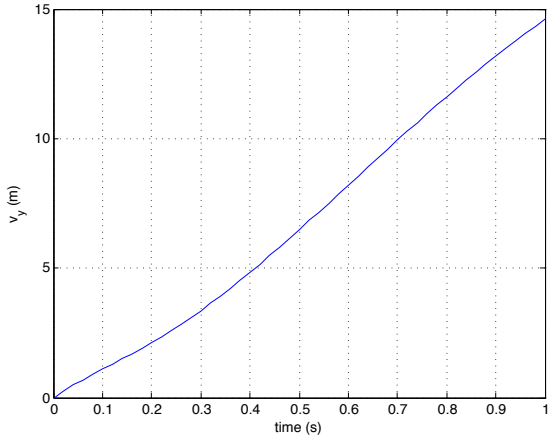

Figure 3.8: Velocity v_y towards the source.

The dynamics of the Renault are considered for an instantaneous road excitation of 10 cm height while driving at a constant forward speed of 72 km/h. The azimuth angle between the satellite signal and the velocity vector of the car is zero. The DOA angle for an array mounted to a vehicle with zero pitch is called $\theta_{\alpha_p=0}$. The new DOA angle in case of pitching can be found by subtracting α_p from $\theta_{\alpha_p=0}$. Note that this DOA calculation assumes that the array center is right above the car's center of mass. The described situation can be seen in figure 3.9.

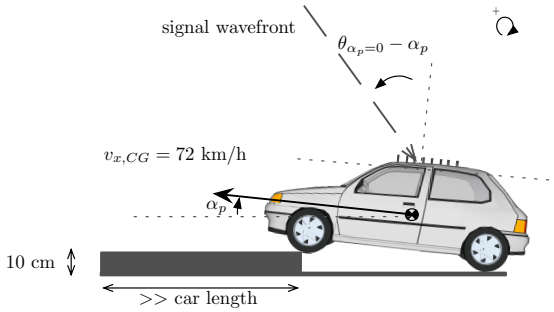

Figure 3.9: DOA calculation angle during sudden road excitation.

If the front wheel hits the excitation, the tire and suspension transfer momentum to the car body. The body starts pitching and this directly influences the DOA of the received signal. Thereafter at approximately a car length distance ($d_f + d_r$ meter) the rear wheel hits the excitation and again momentum is transferred to the car body. The pitch angle dynamics (over time) during the sudden road excitation scenario can be seen in figure 3.10. The forward velocity $v_{x,CG}$ towards the satellite is a constant 72 km/h, thus Doppler effects also occur in this scenario.

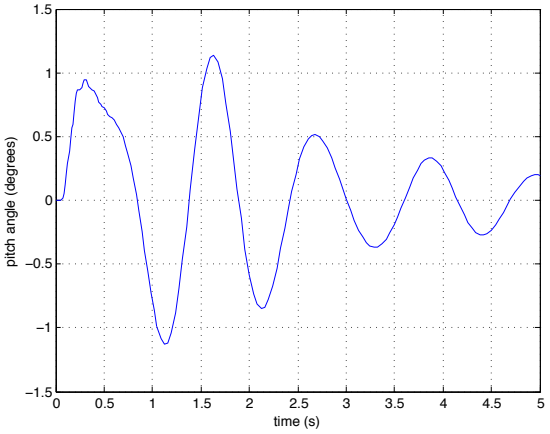

Figure 3.10: Pitch angle α_p over time.

3.4 Conclusion

Given the array configuration described in this work (figure 3.1) the DOA angle of the received signal is only affected by pitching and yawing motion of the car. Two common methods to model yawing and pitching are the planar bicycle model and the half-car suspension model. Based on these models two driving scenarios are simulated to generate DOA and Doppler data that is used for antenna data generation. This antenna data is deployed to test the adaptive steering algorithms. The effects of the 'sudden change in steering angle' scenario on the DOA angle are much larger than the DOA angle changes during the 'sudden road excitation' scenario. Therefore, only the 'sudden change in steering angle' scenario is used in the simulations of chapter 6.

4

Digital Video Broadcasting Satellite

The DVB-S standard describes modulation and channel coding systems for digital satellite reception [ETS97]. The standard allows hardware manufacturers to create equipment that is fully compatible with equipment from other manufacturers and fully compatible with signal broadcasts. Insight in DVB-S modulation and demodulation is necessary for the design and implementation of phased array signal tracking techniques. Demodulation is more intuitive after an introduction of its modulation counterpart. Therefore, this chapter starts by describing the modulation chain in a DVB-S transmitter according to the standard specification [ETS97].

Following upon the introduction of DVB-S modulation, the effects of array movement on the beamformer output are discussed. Based on those effects the blind beamforming algorithms dynamically adapt the array steering vector. An overview of blind beamforming algorithms and their integration in the DVB-S demodulation chain are given in chapter 5.

4.1 DVB-S modulation

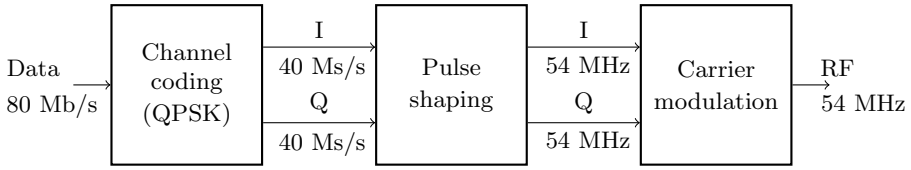

Figure 4.1: DVB-S baseband shaping and modulation.

Figure 4.1 shows the modulation chain of a DVB-S sender. Apart from baseband shaping and modulation the DVB-S specification also describes error control methods used in the transmission system. This report only mentions the channel coding, pulse shaping and carrier modulation parts of the transmission system. Error control is not mentioned because the structural properties of interest of a DVB-S signal (for ex. the constant modulus) are already lost when error control methods are applied.

Channel coding

The sender uses QPSK to map digital input to analog waveforms. QPSK conveys information by changing the phase of the output signal, it uses four different

phases to represent the transmitted information. Every two bits are mapped to symbols that consist of both an In-phase (I) and Quadrature-phase (Q) part. The symbol mapping of QPSK in DVB-S is Gray coded. Gray coding is a coding method where the most likely errors cause only one bit error [Pro01]. The QPSK symbols can be drawn in a constellation diagram that expresses the assignment of bit patterns to specific output values [Com08]. The constellation diagram of Gray coded QPSK can be seen in figure 4.2.

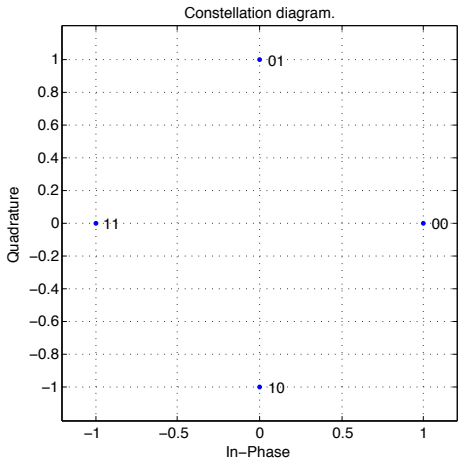

Figure 4.2: Gray-coded QPSK constellation diagram.

The actual DVB-S implementation uses a slightly different version of the described QPSK coding. The constellation diagram is obtained by introducing a $\pi/4$ phaseshift for each symbol from the original QPSK diagram. The extra phaseshift is used by the symbol synchronization at the receiver [Pro01]. In this report the ordinary QPSK constellation without $\pi/4$ extra phaseshift is used, the extra phaseshift has no influence on the simulation results.

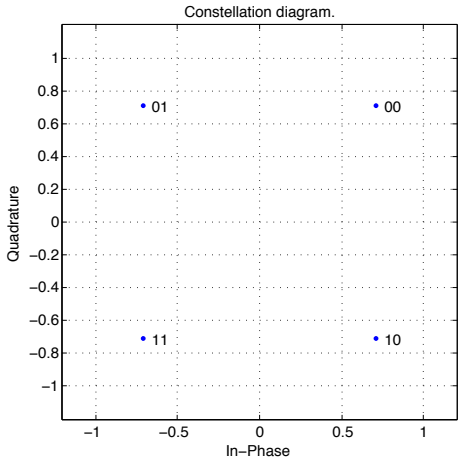

Figure 4.3: Gray-coded QPSK constellation diagram with $\pi/4$ offset.

Pulse shaping

The QPSK coded signal contains abrupt phase changes. Modulating the carrier using this signal creates high frequency components in the output signal, as can be seen in figure 4.4. Therefore, to limit the bandwidth of the output signal and to minimize interference between subsequent symbols pulse shaping is used. Interference between subsequent symbols is also known as Intersymbol Interference (ISI).

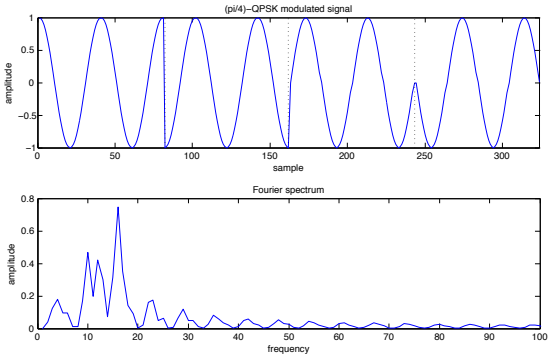

Figure 4.4: Gray-coded QPSK signal.

Pulse shaping is performed by a pulse shape filter. A widely used pulse shape filter is the raised cosine filter. Mathematically the impulse response of a raised cosine filter is given by [Pro01]:

$$p(t) = \frac{\sin(\pi t/T)}{\pi t/T} \frac{\cos(\pi \beta t/T)}{1 - 4\beta^2 t^2/T^2} \tag{4.1}$$

Herein, T is the symbol time and β the roll-off factor which determines the bandwidth of the pulse in the Fourier domain and the filter decay in the time domain.

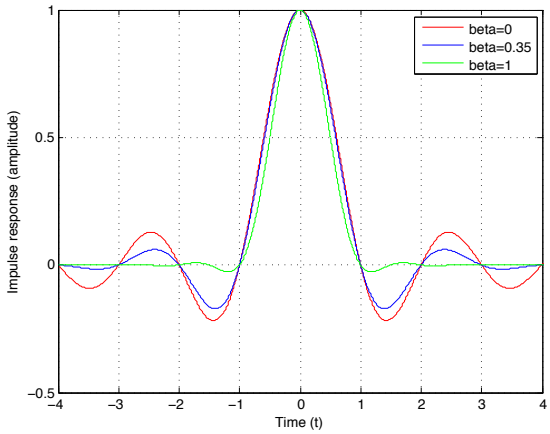

Figure 4.5: Impulse response $p(t)$ of the raised cosine filter.

The filter characteristic becomes more clear by looking at the impulse response $p(t)$ in figure 4.5. Herein, the symbol time T is equal to one and different values for β are used. Greater values of β lead to faster filter decays. The interference between subsequent symbols is minimized because the value of $p(t)$ is zero at the symbol moments ($T = [.. -2, -1, 1, 2..]$).

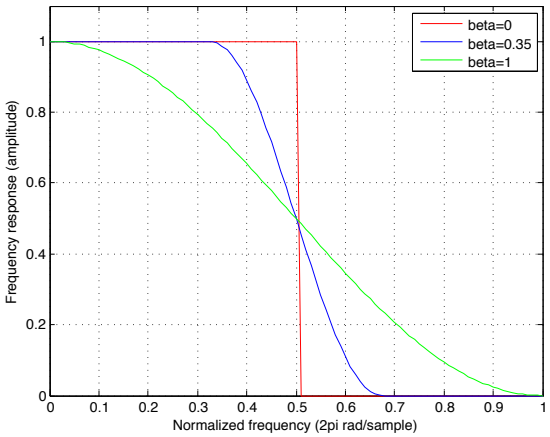

Figure 4.6: Frequency response of the raised cosine filter.

The frequency response of the raised cosine filter is shown in figure 4.6. The normalized frequency of 0.5 corresponds to the Nyquist frequency. It can also be seen that the roll-off factor β determines the bandwidth of the filter. A smaller value of β leads to a smaller bandwidth, but at the cost of a slower filter decay in time. The relation between the symbol rate R_s and the minimum required bandwidth B for a QPSK modulated signal using a raised cosine roll-off β is given by [KHJ05]:

$$R_s = \frac{B}{1+\beta} \tag{4.2}$$

The bit rate of the data entering the DVB-S channel coder is 80 Mbit/s. Every two bits make up one QPSK symbol. Each QPSK symbol consists of an I and a Q value. Therefore, the QPSK encoder has two outgoing signals, both having a sample rate of 40 Msamples per second (Ms/s). Pulse shaping is applied to this data to prevent ISI and to limit the output bandwidth.

The symbol rate R_s of a DVB-S signal is close to 40M symbols per second (Mbaud) [ETS97]. The required bandwidth based on this symbol rate and a roll-off factor 0.35 is $R_s(1+\beta) = 40 \cdot 1.35 \approx 54$ MHz. Each signal entering the pulse shape filter should be filtered to attain the required bandwidth of 54 MHz.

The raised cosine filter can be implemented as a FIR filter. The useful frequency response of a FIR filter is limited to π radians per sample, also known as the Nyquist frequency. Therefore, pulse shaping must operate at a sample rate of twice the original data rate [Gen02]. Thus, the original data should be upsampled by at least a factor two before pulse shaping.

A FIR implementation of the raised cosine filter designed for a 54 MHz bandwidth should operate at a minimal sample rate of twice this bandwidth,

thus at least at 108M samples per second. Therefore the I and Q signal with original samplerates of 40Ms/s per second should both be upsampled to at least 108Ms/s per second before pulse shaping. The results of upsampling and pulse shaping of a QPSK signal can be seen in figure 4.7.

DVB-S performs pulse shaping using a Root-raised Cosine (RRC) type matched filter pair [ETS97]. The receiving filter is matched to the transmitter filter to acquire the desired frequency response of the raised cosine filter. A RRC matched filter pair maximizes the Signal-to-Noise Ratio (SNR) and lowers the ISI [YC05]. Methods for calculating RRC filter coefficients are described in [YC05].

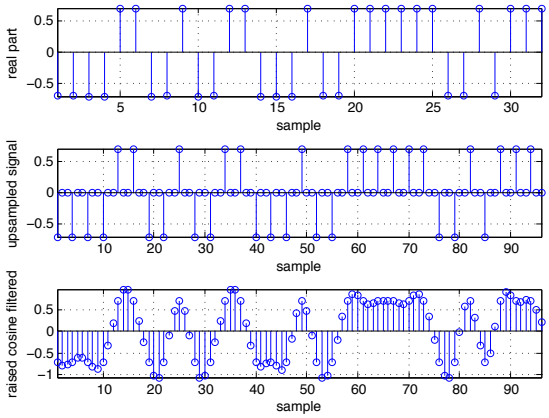

Figure 4.7: Pulse shaping and upsampling of a QPSK signal.

Carrier modulation

The last part of the DVB-S modulation chain is carrier modulation. The I and Q signals are multiplied with a sine and cosine signal oscillating at the carrier frequency to produce the modulated signal at the desired carrier frequency [Com08].

4.2 DVB-S beamforming and demodulation

An introduction to DVB-S modulation and baseband shaping by the sender has been given in the previous section. The signal processing chain for a system that combines beamforming and DVB-S demodulation can be seen in figure 4.8.

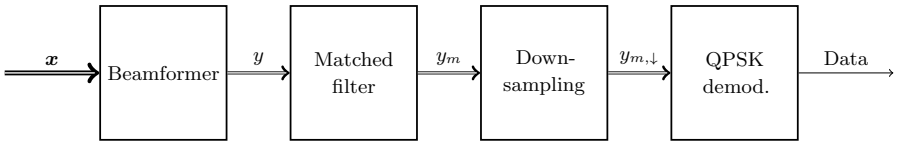

Figure 4.8: Beamforming and demodulation of DVB-S signals.

The input of the system is a vector of N complex samples coming from the phased array antenna front-ends, this vector is indicated by $\boldsymbol{x}$. Vector $\boldsymbol{x}$ is processed by the beamformer block which uses phaseshift based beamforming to steer the directivity of the array.

After beamforming a matched filter of the root raised cosine type is used. The samplerates of both $\boldsymbol{x}$ and y are chosen to be an integer multiple of the DVB-S symbol rate. Section 4.1 explained that at least 108 Ms/s is required, therefore the samplerates of $\boldsymbol{x}$ and y are set to three times the DVB-S symbolrate ($3 \cdot 40 = 120$ Ms/s).

Eventually, the root raised cosine filtered signal is downsampled to the DVB-S symbol rate (40 Ms/s) before it is QPSK demodulated to retrieve the transmitted data symbols.

4.3 DVB-S beamforming in dynamic environements

The beamformer output is affected by movements of the array. The effects of array movement can be recognized in scatter plots of the beamformer output. A scatter plot shows the modulated signal symbols as dots in an I and Q diagram. Ideally, those dots are exactly at the position of one of the constellation points of the used modulation technique. However, noise introduces deviation of the scatter dots from those constellation positions. In figure 4.9 the scatter plot is shown of QPSK reception in clean air without beam mispointing. Signal reception in clean air corresponds to a SNR of 16 dB [PBA03].

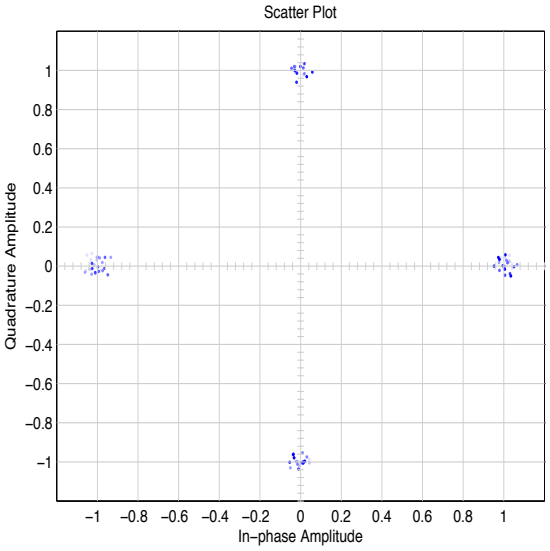

Figure 4.9: Beamformer output without mispointing (16 dB SNR).

A changing DOA results in beam mispointing if there is no correction of the array steering vector in the direction of the new DOA. In this report two types of movement are discussed:

- Translation: every point of the array moves in the same direction.

- Rotation: the array rotates around a specified pivot point.

Translation and rotation cause different phase and magnitude effects in the received antenna signals $\boldsymbol{x}$ and the beamformer result y. Those effects will be discussed in the upcoming sections. Additionally, the relation between the phase reference of the array and the array phase transfer will be shown.

Translational array dynamics

Translational movement of a phased array can be seen in figure 4.10. Herein, a four element antenna array is shown translating from position P to P'. The array moves closer to the transmitter, thus the difference in signal path length ($L \neq L'$) leads to a phaseshift. The amount of phaseshift is the same for all antenna elements and depends on the Direction of Arrival (DOA) angle θ and the direction of movement. This effect is better known as the Doppler effect. Note that the Direction of Arrival (DOA) does not change ($\theta = \theta'$) while translating, for that reason beamsteering is not required in case of translational movement.

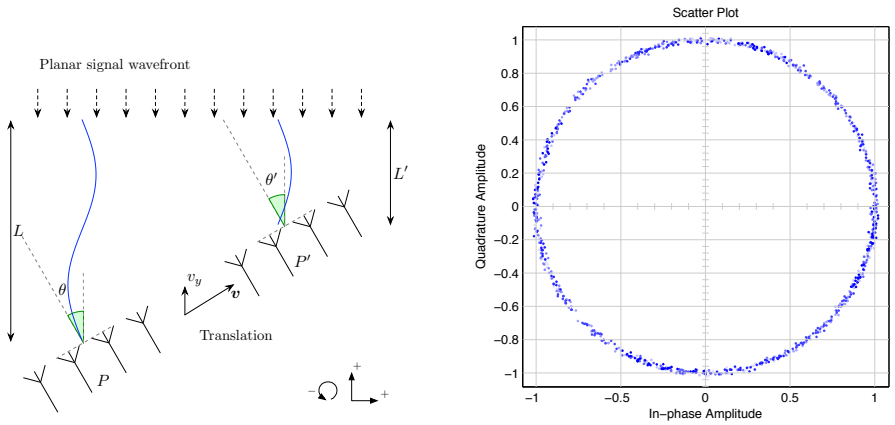

Figure 4.10: Array translation.

Mathematically the phase shift changes $\dot{\varphi}_D$ (in radians per second) of the antenna signals can be given as function of the signal wavelength and the array velocity in the direction of the planar wavefront (in meters per second):

$$\dot{\varphi}_D = v_y \cdot \frac{2\pi}{\lambda} \tag{4.3}$$

Herein, λ is the wavelength of the received signal in meters. Translation orthogonal to v_y does not introduce a Doppler phaseshift. An equally sized phaseshift of all the antenna signals $\boldsymbol{x}$ leads to a phase offset in the beamformer result. A moving array results in a changing phase offset. This effect can be recognized in the rightmost graph of figure 4.10. The rotation of the scatter points is caused by Doppler phaseshift.

Rotational array dynamics

The phase and magnitude transfer of a rotating array depend on the phase reference of the array. First the phase and magnitude effects will be discussed for the situation where the phase reference is at the center of the array. Thereafter the effects are described for the case where it is not centered.

Centered phase reference and pivot point

Phase changes in the beamformer output caused by rotation can be eliminated whenever the center of the array is chosen as the phase reference, this situation is shown in the leftmost graph of figure 4.11. Herein, $\phi(\theta)$ is defined as the phase delay between two subsequent antenna elements for a DOA angle θ and signal wavelength λ.

$$\phi(\theta) = 2\pi \left(\frac{d \cdot \sin(\theta)}{\lambda} \right) \tag{4.4}$$

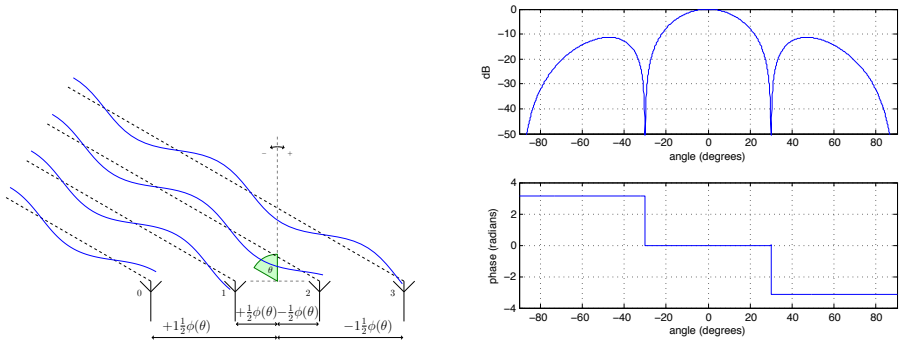

Figure 4.11: The array center is chosen as the phase reference.

The array pattern of a four element ULA with a phase reference in the center is shown in rightmost graph of figure 4.11. The gain of the array is decreasing in case of small mispointing angles. For small mispointing angles the phase of the output signal does not change. However, after a null the output phase gets a 180 degrees shift.
A scatter plot of the four element ULA in case of a -13 degree mispoint can be seen in figure 4.12. The array pattern of the ULA indicates that -13 degrees corresponds to -3 dB gain. The amplitude A of the received signal in figure 4.12 is equal to 0.7, which corresponds to half the transmitted power ($P = A^2 = 0.7^2 \approx 0.5$). Notice that a mispoint of a positive 13 degrees corresponds to the same -3 dB gain and leads to a similar scatter plot.

Off-center phase reference and pivot point

Analogue to the previous section the effects caused by an off-center phase reference will be explained by looking at the gain and phase transfer of a four element ULA. However, this time the phase reference is located at the first antenna element on the right side of the center of the array. The described array configuration can be seen in leftmost graph of figure 4.13.

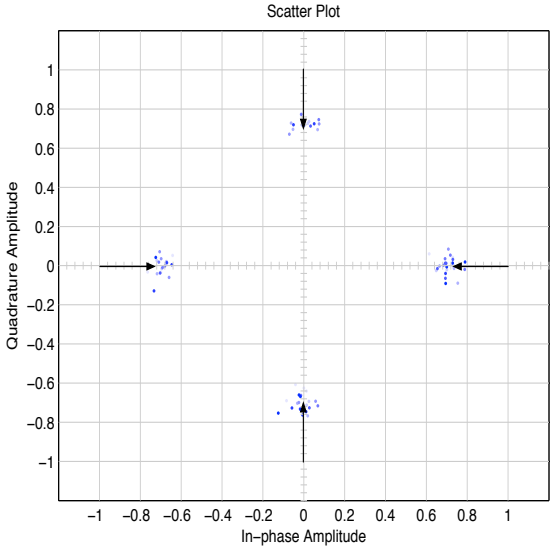

Figure 4.12: Modulus decrease due to mispointing.

The phase and gain transfer of this array configuration can be seen in rightmost graph of figure 4.13. The off-center phase reference introduces phase changes for small mispointing angles. Those phase changes can clearly be recognized in a scatter plot of the beamformer output. Figure 4.14 shows a scatter plot of the beamformer output of the array (with an off-center phase reference) for a -15 degrees mispoint. The negative arrival angle leads to both a modulus decrease and a negative phaseshift of the beamformer result.

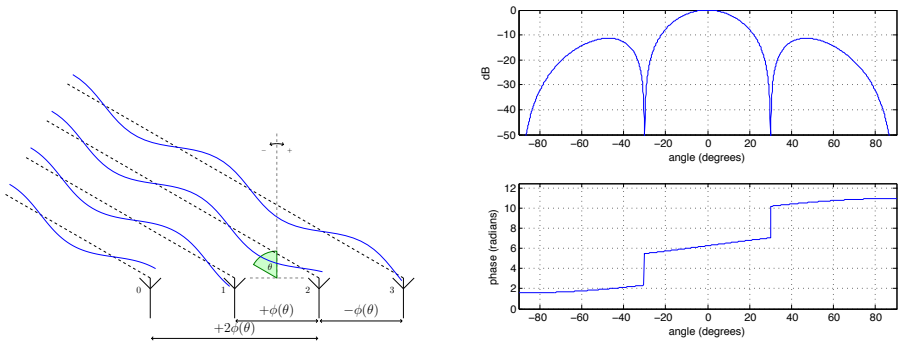

Figure 4.13: Off-center phase reference.

Due to the occurence of both a phaseshift and a modulus decrease whenever the phase reference is not centered, the effects of translational and rotational dynamics are hard to distinguish. Therefore, a centered phase reference is preferred in this work.

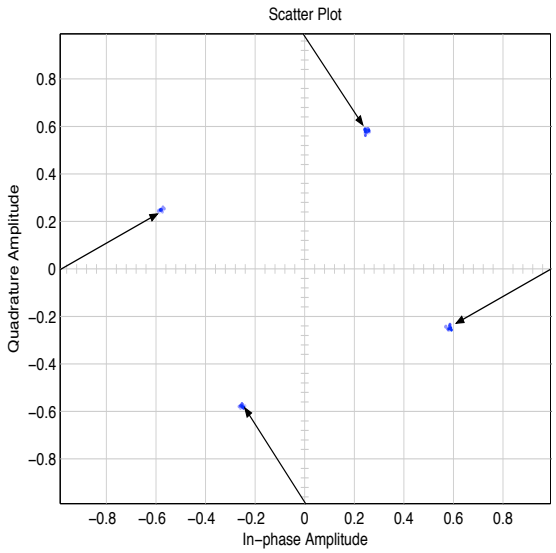

Figure 4.14: Modulus decrease and phase error due to mispointing.

4.4 Conclusion

DVB-S signals are QPSK modulated. QPSK conveys information by changing the phase of the output signal. Four different phases are used to represent the symbols to be transmitted. Before the carrier signal is modulated by the QPSK coded signal pulse shaping is performed to limit the bandwidth and minimize the ISI of the output signal. A matched (pulse shape) filter is used in the receiver chain, it requires a certain minimum sample rate. Determination of this minimum sample rate is explained in section 4.1. Beamforming in the receiver chain is performed before matched filtering, thus the minimum samplerate of the beamformer is imposed by this matched filter sample rate.

The effects of array movement on the beamformer output depend on the phase reference of the array. If the phase reference is centered then translational and rotational movement lead to orthogonal effects. Translational movement causes rotation of symbols in the scatter plot, because all the antenna signals get an equal phase shift. Rotational movement leads to beam mispointing and results in a modulus decrease of the scatter points. For an off-center phase reference both effects are not orthogonal anymore.

5

Blind beamforming of DVB-S signals

The dynamic behaviour of the DOA (as seen from the array) introduces modulus and phase deviations in the QPSK modulated signal. A blind beamforming algorithm uses those deviations to adapt the array steering vector to prevent it from mispointing. Two adaptive algorithms are introduced that can be used for blind beamforming of DVB-S signals, the Constant Modulus Algorithm (CMA) and an extended version of CMA.

5.1 Constant Modulus Algorithm

Section 4.3 mentioned the modulus decrease of scatter points as an effect of mispointing. If the zero-phase reference is positioned in the center of the array then mispointing only affects the modulus of the beamformer output. Ideally, this modulus (after matched filtering and downsampling) is equal to one.
One of the possible adaptive array algorithms that copes with mispointing is the CMA algorithm. The CMA algorithm changes the steering vector $\boldsymbol{\phi}$ until the modulus is constantly equal to one [vdV04].

CMA is based on a cost function and gradient descent methods to decrease costs. Both topics are discussed in the following sections.

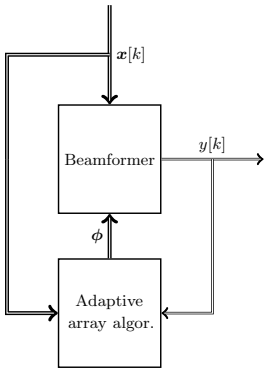

Figure 5.1: General form of an adaptive beamformer.

Cost function

The cost function is defined as the expected deviation of the squared modulus of the beamformer output with respect to a constant value. For a QPSK

modulated signal this constant value is one, since all the QPSK symbols lie on the unit circle. Based on the notation of figure 5.1 the CMA cost function is mathematically given as [vdV04]:

$$J(\boldsymbol{\phi}) = E(|y[k]|^2 - 1)^2 = E(\left|\boldsymbol{\phi}^H \boldsymbol{x}[k]\right|^2 - 1)^2 \tag{5.1}$$

The aim of CMA is to minimize $J(\boldsymbol{\phi})$ by altering $\boldsymbol{\phi}$. Lower costs means less deviation from the constant modulus. The cost function gets more intuitive when it is drawn in a surface plot. In figure 5.2 the x-axis shows the real part of the beamformer output $y[k]$, the y-axis shows the imaginary part of this output and the z-axis shows the corresponding costs J. Minimum costs are reached whenever the values of $y[k]$ have a constant modulus of one, thus whenever those values are lying on the unit circle.

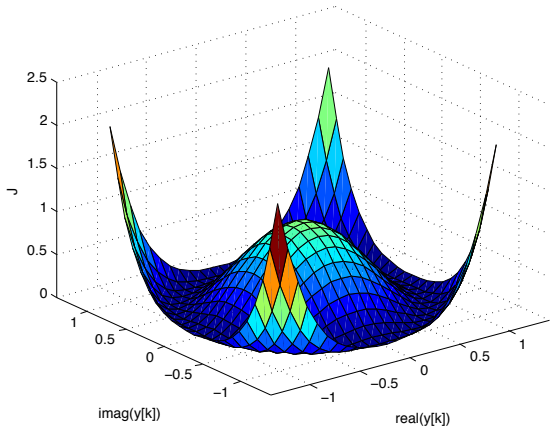

Figure 5.2: Surface plot of the CMA cost function.

Gradient descent optimization technique

Minimization of the CMA cost function can be done using many different techniques [vdV04]. The simplest method is to iteratively minimize J using a stochastic gradient-descent. This means that the gradient $\nabla_{\boldsymbol{\phi}} J$ is calculated and the steering vector $\boldsymbol{\phi}$ is updated in the direction of the negative gradient to mimize J. Mathematically this can be written as [TA83]:

$$\boldsymbol{\phi}[n+1] = \boldsymbol{\phi}[n] - \mu \nabla_{\boldsymbol{\phi}} J \tag{5.2}$$

Herein, μ determines the convergence rate and can be determined during simulation. Note that n instead of k is used to index steering vector updates. This is done to indicate that the rate of steering vector updates differs from the beamformer samplerate. The ratio of the beamformer sample rate over the steering vector update rate can be expressed by R:

$$R = \frac{\text{beamformer sample rate}}{\text{steering vector update rate}} \tag{5.3}$$

At every R^{th} sample processed by the beamformer the steering vector is updated.

CMA based on gradient descent can be explained as follows. The beamformer outputs a sample $y[k]$, this sample value lies somewhere on the surface of the cost function shown in figure 5.2. Every R^{th} sample of the beamformer output CMA outputs a new array steering vector $\boldsymbol{\phi}$ that leads to the steepest descent of the cost function J to the unit circle minimum.

Calculation of the steering vector $\boldsymbol{\phi}$ requires an expression for the gradient of J. The gradient can be derived as followed. At first the chain rule is applied to J and the beamformer output is rewritten using $y[k] = \boldsymbol{\phi}^H \boldsymbol{x}[k]$. This expression for beamforming output was already discussed in section 2.15:

$$\begin{aligned} J &= E(|y[k]|^2 - 1)^2 \\ \nabla_{\boldsymbol{\phi}} J &= E\{2 \cdot (|y[k]|^2 - 1) \cdot \nabla_{\boldsymbol{\phi}} (|y[k]|^2 - 1)\} \\ \nabla_{\boldsymbol{\phi}} J &= 2 \cdot E\{(|y[k]|^2 - 1) \cdot \nabla_{\boldsymbol{\phi}} (\left|\boldsymbol{\phi}^H \boldsymbol{x}[k]\right|^2 - 1)\} \end{aligned} \tag{5.4}$$

The term $\left|\boldsymbol{\phi}^H \boldsymbol{x}[k]\right|^2$ can be written as $\boldsymbol{\phi}^H \boldsymbol{x}[k] \boldsymbol{x}[k]^H \boldsymbol{\phi}$ [vdV04]. Thereafter, application of the product rule gives:

$$\begin{aligned} \nabla_{\boldsymbol{\phi}} J &= 2E\{(|y[k]|^2 - 1) \cdot \nabla_{\boldsymbol{\phi}} (\boldsymbol{\phi}^H \boldsymbol{x}[k] \boldsymbol{x}[k]^H \boldsymbol{\phi} - 1)\} \\ \nabla_{\boldsymbol{\phi}} J &= 4E\{(|y[k]|^2 - 1) \cdot \boldsymbol{x}[k] \boldsymbol{x}[k]^H \boldsymbol{\phi}\} \\ \nabla_{\boldsymbol{\phi}} J &= 4E\{(|y[k]|^2 - 1) \cdot y[k]^* \boldsymbol{x}[k]\} \end{aligned} \tag{5.5}$$

A combination of the minimizer from equation 5.2 and the result of the gradient derivation of J gives the recurrence relation for the (gradient descent) minimizer [TA83]:

$$\boldsymbol{\phi}[n+1] = \boldsymbol{\phi}[n] - \mu \cdot 4\{(|y[k]|^2 - 1) \cdot y[k]^* \boldsymbol{x}[k]\} \tag{5.6}$$

In literature the factor 4 is often absorbed in the convergence rate μ. Absorbance of this factor in μ and rewriting the recurrence relation in causal form gives the following equation for the CMA minimizer:

$$\boldsymbol{\phi}[n] = \boldsymbol{\phi}[n-1] - \mu \cdot \{(|y[k-1]|^2 - 1) \cdot y[k-1]^* \boldsymbol{x}[k-1]\} \tag{5.7}$$

Equation 5.7 is shown in figure 5.3. This block diagram enables clear identification of the different operations and signal types.

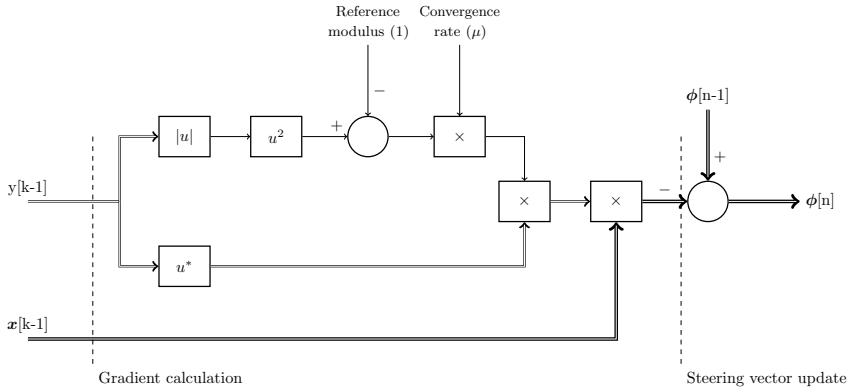

Figure 5.3: Block diagram of the CMA algorithm.

5.2 Extending the CMA cost function

In the previous section the CMA algorithm was introduced. CMA adjusts steering vector weights based on modulus effects in the beamformer output caused by mispointing. Section 4.3 mentions that translational array dynamics cause phase shift changes in the beamformer output. A moving vehicle experiences both translational and rotational movement, thus phase deviations in the beamformer output are unavoidable. Therefore, CMA for blind beamforming on a mobile phased array can be improved by adjusting the steering vector weights based on both the modulus and phase information from the beamformer output.

The idea of using phase information for equalization of QPSK signals is (first) mentioned in [Zhe00]. It can be implemented by adding another constraint to the CMA cost function. However, in [Zhe00] this is not directly applied in a blind beamforming context. The instantaneous phase φ of a complex signal $y = y_I + y_Q \cdot j$ can be found as followed:

$$y_\varphi = \arctan\left(\frac{y_Q}{y_I}\right) = \arctan\left(\frac{y - y^*}{j(y + y^*)}\right) \tag{5.8}$$

The constellation diagram of a QPSK modulated signal consists of symbols with phases equally distributed on a circle with constant modulus. This equal phase distribution of a QPSK modulated signal y can be mathematically expressed by $\sin(2y_\varphi) = 0$. The phase part of the new cost function is defined as the expected squared deviation of $\sin(2y_\varphi)$ from zero:

$$J = E\left(\left(\sin\left(2y_\varphi\right) - 0\right)^2\right) = E\left(\sin^2\left(2y_\varphi\right)\right) \tag{5.9}$$

Combining the phase cost function and the CMA cost function mentioned in equation 5.1 leads to the following expression for the extended CMA cost function [Zhe00]:

$$\begin{aligned} J &= E\left(\left|y[k]\right|^2 - 1\right)^2 + E\left(\sin^2\left(2y_\varphi[k]\right)\right) \\ J &= E\left(\left|y[k]\right|^2 - 1\right)^2 + E\left(\sin^2\left(2 \cdot \arctan\left(\tfrac{y[k]-y[k]^*}{j(y[k]+y[k]^*)}\right)\right)\right) \end{aligned} \tag{5.10}$$

Herein, $y[k]$ is the beamformer output signal. The extendend CMA cost function assumes that modulus and phase deviations are weighted equally. The cost function is more intuitive if it is drawn in a surface plot like the CMA cost function in figure 5.4. Minimum costs are reached whenever the values of $y[k]$ simultaneously have a unit modulus and a phase equal to one of the QPSK symbol phases. Deviation from this unit modulus and deviation from those equally distributed phases both lead to cost increase.

The aim of the extended CMA algorithm is minimizing the costs J by playing with the steering vector $\boldsymbol{\phi}$. If expression $y[k] = \boldsymbol{\phi}^H \boldsymbol{x}$ is used to describe beamforming then equation 5.10 can be rewritten as function of $\boldsymbol{\phi}$:

$$J(\boldsymbol{\phi}) = E\left(\left|\boldsymbol{\phi}^H \boldsymbol{x}\right|^2 - 1\right)^2 + E\left(\sin^2\left(2 \cdot \arctan\left(\frac{\boldsymbol{\phi}^H \boldsymbol{x} - \boldsymbol{x}^H \boldsymbol{\phi}}{j\left(\boldsymbol{\phi}^H \boldsymbol{x} + \boldsymbol{x}^H \boldsymbol{\phi}\right)}\right)\right)\right) \tag{5.11}$$

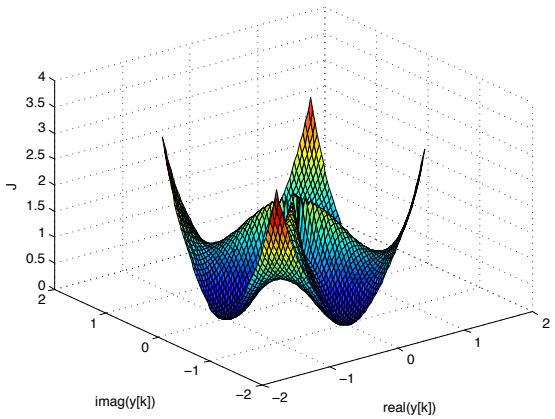

Figure 5.4: Surface plot of the extended CMA cost function.

Gradient descent solution

Minimization of cost function 5.11 is performed by the same gradient-descent technique which was used for the CMA algorithm of section 5.1. The gradient $\nabla_{\boldsymbol{\phi}} J$ is calculated and used in the gradient descent recurrence relation (equation 5.2).
The derivation of $\nabla_{\boldsymbol{\phi}} J$ is not given here. It basically consists of steps similair to the ones of section 5.1, however the gradient derivation of equation 5.11 is a bit more involved. A full derivation can be found in [Zhe00]. The following recurrence relation is obtained for the CMA minimizer if it incorporates phase deviation costs:

$$\boldsymbol{\phi}[n+1] = \boldsymbol{\phi}[n] - \mu \cdot \frac{8j\left(|y[k]|^4 - |y[k]|^2\right) + 4\sin\left(4y_{\varphi}[k]\right)}{4j \cdot y[k]} \cdot \boldsymbol{x} \qquad (5.12)$$

Herein, $y[k]$ is the beamformer output and $y_{\varphi}[k]$ is calculated using equation 5.8.

5.3 DVB-S blind beamforming

Figure 5.5 shows the block diagram of figure 4.8 extended with an adaptive blind beamformer. The minimizer in the adaptive algorithm block calculates steering vector updates based on a causal version of minimizer equation 5.6 or 5.12. Those equations require antenna snapshot $\boldsymbol{x}[k]$ and the beamformer output $y[k]$ as input. In figure 4.8 it is shown that the 'original' transmitted data symbols are obtained after matched filtering and down sampling. The minimizer equation uses this (matched filtered and down sampled) beamformer output $y_{m,\downarrow}$ as input. Thus, the antenna snapshot $\boldsymbol{x}[k]$ should also be delayed and down sampled before it is provided to the minimizer in the adaptive algorithm block.

The sample rates of the beamformer block and the adaptive array algorithm block differ because at the latter inputs are down sampled. This sample rate difference is also noted in section 5.1. A sample and hold circuit at the adaptive

algorithm block output connects both sample rate domains. A more in-depth discussion of this blind beamforming system can be found in chapter 6.

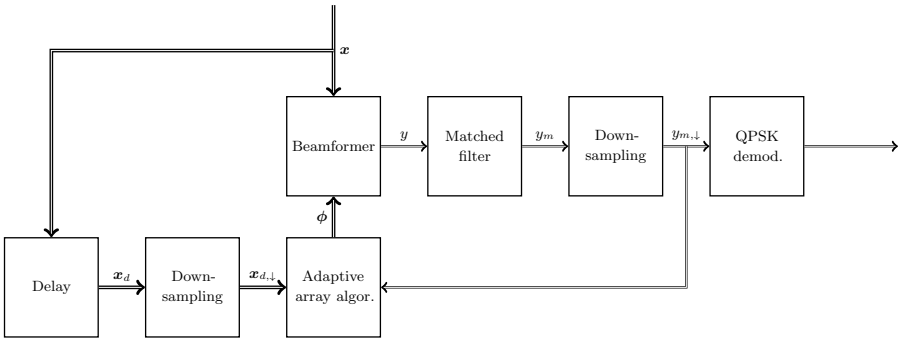

Figure 5.5: Blind beamforming of DVB-S signals.

5.4 Conclusion

Two adaptive algorithms are discussed that can be used for blind beamforming of DVB-S signals, the CMA algorithm and an extended version of CMA. Both algorithms define a cost function J that is minimized using a gradient descent.

The CMA cost function is based on deviations of the beamformer output from a constant modulus. The extended CMA cost function is based on the same modulus deviations, but extended to also include phase deviations of the beamformer output from QPSK symbol phases.

In a moving vehicle phase deviations are unavoidable due to translational movement. Therefore, the extended CMA algorithm is expected to outperform the original CMA algorithm. Simulations of both algorithms are performed in chapter 6.

6

Modelling and simulation

The previous chapter introduced multiple algorithms for adaptive steering of phased array weights. Those algorithms need to be verified against the dynamic scenarios mentioned in chapter 3. The simulations are performed to gather conclusions about the usability of the algorithms. Furthermore, some initial timing requirements can be recognized which are necessary for building hardware implementations of the algorithms.

In section 6.1 an overview of the simulation model and its parameters is given. Section 6.2 describes the simulation results of both CMA and the extended version of CMA applied in dynamic environments. Section 6.3 mentions timing and update frequency requirements obtained from the simulations. These requirements are followed by short conclusions on the results of this chapter.

6.1 Simulation overview

Modelling and simulation of the adaptive beamforming system are performed using the MathWorks^{TM}Matlab and Simulink technical computing software. A block diagram of the simulation model can be seen in figure 6.1. The simulation can be divided in four separate parts:

- Signal generation
- Beamforming
- DVB-S demodulation
- Adaptive steering

The signal generation subsystem creates a random QPSK modulated signal. The transmit angle θ and Doppler effects of this signal are determined by one of the dynamic vehicle models mentioned in chapter 3. The antenna data is generated using the channel model of equation 2.6. The result of the signal generation is the complex antenna snapshot $\boldsymbol{x}[k]$ that represents the complex samples of all antenna elements at time instant k. The sample rate of $\boldsymbol{x}[k]$ is set to 120M complex samples per second to enable simulation of realistic DVB-S bandwidths with pulse shaping and upsampling as explained in section 4.1. Pulse shaping is performed by a RRC filter implemented using a 25-tap FIR filter. The use of 25-taps results in a smooth band-pass frequency response.

The antenna snapshot $\boldsymbol{x}[k]$ is input to the beamformer. The beamformer output $y[k]$ is calculated using equation $y[k] = \boldsymbol{\phi[k]}^H \boldsymbol{x}[k]$. The formula for

beamforming described in equation 2.13 can also be used, but this requires complex conjugation of the adaptive array algorithm output.

The beamforming result $y[k]$ is input for the DVB-S demodulation and the adaptive steering subsystems. The DVB-S demodulation subsystem starts by filtering the beamformer output $y[k]$ using a 25-tap RRC matched FIR filter. The result of the matched filter $y_m[k]$ is an upsampled version of the original QPSK modulated signal. Downsampling is applied to decrease the sample rate from 120M samples per second to the original rate of 40M samples per second ($R = 3$). Finally, this downsampled signal is input for the QPSK demodulator that matches its input samples to QPSK constellation points.

Beam pointing and suppression of interferers is performed by the adaptive steering subsystem. In chapter 5 it was shown in figure 5.1 that an adaptive beamformer requires two input signals: the antenna snapshot $\boldsymbol{x}[k]$ and the beamforming result $y[k]$. The adaptive subsystem in figure 6.1 operates on down-sampled versions of the these signals, respectively $y_{m,\downarrow}$ and $\boldsymbol{x}_{d,\downarrow}$. The delay block before downsampling is required to synchronize $y_{m,\downarrow}$ and $\boldsymbol{x}_{d,\downarrow}$. The delay should be equal to the group delay of the matched filter, in case of a 25-tap matched FIR filter a delay of 12 sample periods is introduced.

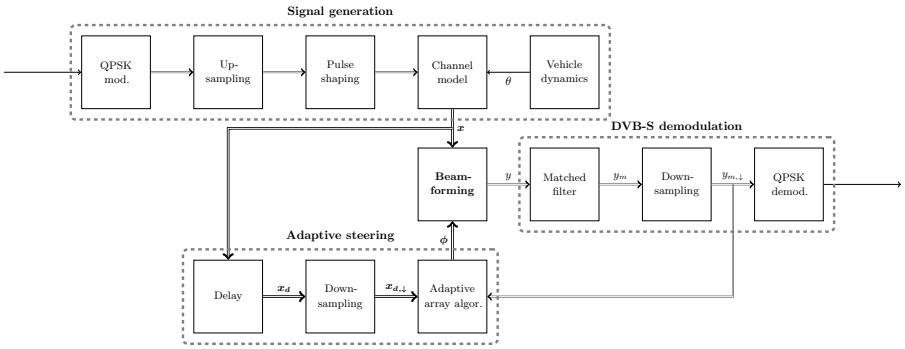

Figure 6.1: Simulation model for DVB-S blind beamforming.

6.2 Simulation results

The simulation results of the blind beamformer model (figure 6.1) are discussed in the upcoming sections. In the simulations the original Constant Modulus Algorithm (CMA) as well as the version with an extended cost function are used in the adaptive steering subsystem.

Adaptive steering performance

The two steering algorithms are tested in two different scenarios. The first scenario is a synthetic scenario which introduces the methods of performance analysis and deals with interpretation of the simulation data. Thereafter, the simulation is improved by using the planar bicycle model of chapter 3 for a realistic transmit angle generation.

Sinusoidal steering angle

The sinusoidal steering angle scenario is a synthetic situation meant to introduce the methods of performance analysis. In this scenario the following parameters were used:

Adaptive algorithm	*CMA*
Transmit angle generation	*100 Hz sinus*
	80 degrees peak-to-peak
Number of antenna elements	*8*
Convergence rate μ	*0.005*
Phase reference	*array center*
AWGN SNR	*16 dB*
Interfering sources	*none*

Table 6.1: Parameters for the sinusoidal steering angle simulation.

An analysis of the adaptive algorithms can be based on many different criteria. The three criteria that were used to compare the algorithms from chapter 5 are:

- Radiation pattern dynamics
- (extended) CMA costs
- Symbol Error Rate (SER) dynamics

A plot of multiple radiation patterns (stacked together over time) is useful to visualize the dynamic behaviour of an adaptive algorithm, such a plot can be seen in figure 6.2. Note that at time instant $t = 0$ the radiation pattern from figure 2.2 can be recognized. The black line indicates the Direction of Arrival (DOA) reference.

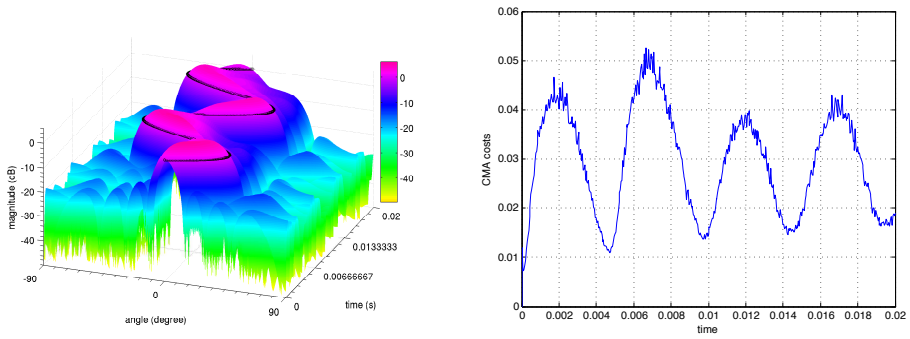

Figure 6.2: Dynamic behaviour of the radiation pattern and the CMA costs.

Based on the radiation pattern dynamics it can be visually concluded that the CMA algorithm is able to update the steering vector weights in such a manner that the array is highly sensitive in the reference direction. If the signal is received properly the CMA costs (equation 5.1) should remain close to zero. The latter can be verified in figure 6.2 that shows the CMA costs over time in the rightmost graph.

A more common method to express the performance of communication channels is to calculate the Signal-to-Noise Ratio (SNR). The SNR implies the Bit Error Rate (BER) and the SER. The SNR in decibel is defined as [HH89]:

$$SNR = 10 \log_{10} \left(\frac{P_s}{P_n} \right) \tag{6.1}$$

Herein, P_s represents the signal power and P_n represents the noise power. The SNR of the adaptive beamformer (figure 6.1) can be determined based on the spectrum of the beamformer output y. During simulation every 256 samples are input to a 256-point FFT. An example of a spectrum can be seen in figure 6.3.

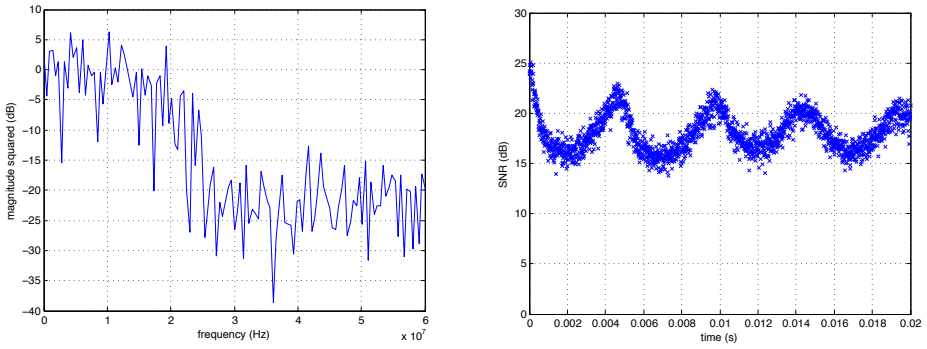

Figure 6.3: Spectrum of the beamformer output and the SNR over time.

The FFT spectrum of the beamformer output shows a clear power level difference between the signal band and the noise floor. During simulation the SNR is calculated of each FFT spectrum by repeatingly executing the following steps:

1. Calculate the mean of the power levels that belong to the signal bandwidth. The frequencies from 0-20 MHz belong to signal band; their mean is calculated.

2. Calculate the mean of the power levels that belong to the noise floor. The frequencies from 30-60 MHz belong to the noise floor; their mean is calulated.

3. Divide both means to determine the Signal-to-Noise Ratio (SNR) of the input spectrum.

In [Pro01] it can be seen that for a Quasi Error Free (QEF) output signal of the error decoding system, a BER of $2 \cdot 10^{-4}$ is required at the output of the QPSK demodulator. For a QPSK signal to be demodulated with a BER of $2 \cdot 10^{-4}$, the minimum required SNR for QPSK demodulation in a DVB-S receiver is 13 dB. The results of the SNR calculations for the adaptive beamformer can be seen in the rightmost graph of figure 6.3. The SNR of the beamformer output stays above 13 dB, therefore the BER is sufficiently low.

In the previous simulation only rotational effects were modelled. However, in a more realistic scenario the array also experiences translational movement.

Translation of the array leads to phase effects in the beamformer output y, as shown in chapter 4.

The CMA algorithm does not compensate for phase changes in the beamformer output. However, the extended CMA does include phase offset correction. Therefore, the extended CMA is expected to outperform the original CMA algorithm for situations where both modulus and phase changes occur. The previous assumption is tested in an extended simulation that includes Doppler effects while other parameters are equal to those in table 6.1. The phase effects in this simulation are modeled by a velocity v_y of 20 meters per second in the direction of the source. Comparison of the algorithms has been done based on the running SER over time and can be seen in figure 6.4.

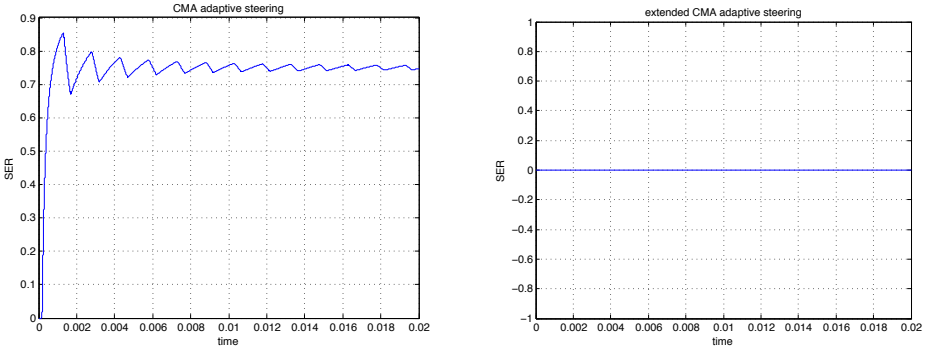

Figure 6.4: SER in synthetic scenario with Doppler effects.

The simulation shows that CMA with phase correction clearly outperforms the classic CMA algorithm for situations with Doppler effects. The SER in the leftmost graph of figure 6.4 eventually turns out to be 0.75, because three quarters of the time the constellation points are in the wrong QPSK quadrant.

Based on the results from figure 6.4 the upcoming simulation only uses the extended version of CMA because both phase and modulus changes occur in the beamformer output.

Sudden change in steering angle

Simulation of the extended version of CMA for the 'sudden change in steering angle' scenario is required to gather conclusions about its performance in a realistic setting. The sudden steering angle scenario is described in chapter 3. The heading angle ψ and the velocity v_y towards the source are now part of the vehicle dynamics block of figure 6.1. An overview of the parameters used for the simulation can be seen in table 6.2.

Dynamic behaviour of the beam pattern when the extended CMA algorithm is used for steering can be seen in figure 6.5. The result clearly shows that an adaptive array mounted on the roof of a vehicle is able to flawlessy track a DVB-S satellite signal during the first second after a sudden change in steering angle. In section 6.3 this scenario is simulated with decreased processing rates for the adaptive steering system to find a lower bound that still provides correct signal tracking.

Adaptive algorithm	*extended CMA*
Transmit angle generation	ψ *from scenario*
Doppler effect generation	v_y *from scenario*
Number of antenna elements	*8*
Convergence rate μ	*0.005*
Phase reference	*array center*
AWGN SNR	*16 dB*
Interfering sources	*none*

Table 6.2: Parameters for the 'sudden change in steering angle' scenario.

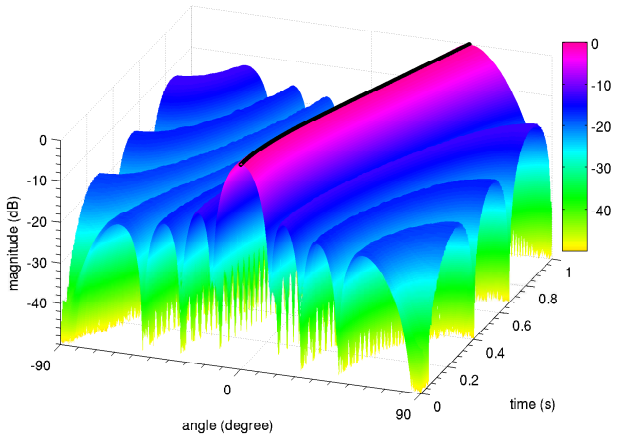

Figure 6.5: Beam pattern for the 'sudden change in steering angle' scenario.

6.3 Update frequency requirements

In the simulation model the adaptive array algorithm processes 40M samples per second. The processing requirements for this rate are considerable. Therefore, it would be benificial for a hardware implementation if the adaptive array algorithm is able to steer the beam at a lower rate. In chapter 5 the ratio of the beamformer sample rate over the steering vector update rate is called R. Running the adaptive array algorithm at a lower rate requires an additional downsampling operation on the signal $y_{m,\downarrow}$ entering the adaptive array algorithm. The required rate of this extra downsampling operation can be found by dividing R by the rate of the downsampling operation in the DVB-S demodulation subsystem.

The extended CMA cost function (equation 5.11) gives an indication of proper modulus and phase correction of the beamformer output signal. The sudden change in steering angle scenario is simulated multiple times for different ratios of R. The mean of the extended CMA costs for each run is shown in figure 6.6. Figure 6.6 clearly shows that after a certain point the costs seem to grow exponentially. Simulations with an $R >= 531$ turn out to be unsuccesfull, because of the high bit error rates.

The convergence parameter μ is not changed during the different runs. A

larger μ could increase the maximum value of R. The parameter μ affects the amount of correction the algorithm is able to perform in each iteration. In a hardware implementation the value of μ should be optimized by parameter optimization.

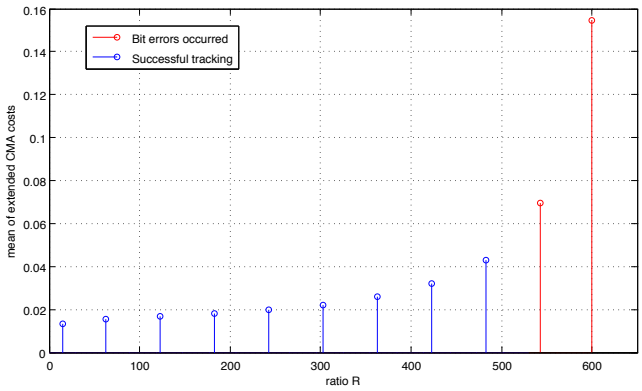

Figure 6.6: Extended CMA costs for different ratios R.

6.4 Conclusion

The performance of the adaptive steering algorithms (chapter 5) for the sudden steering angle scenario (chapter 3) has been tested. CMA and the extended CMA algorithm are both able to correctly update the steering vector weights to steer the array sensitivity in the source direction. Unlike CMA, the extended CMA algorithm can simultaneously correct for Doppler effects.

Initially, the update rate of the adaptive algorithm block was set to 40M samples per second. This corresponds to a ratio of the beamformer sample rate over the steering vector update rate of three. Simulations have shown that this ratio R can be increased to more than five hundred without affecting the number of bit errors.

7

Analysis of the blind adaptive algorithms

The previous chapter discusses simulation of CMA and the extended CMA algorithm for different dynamic scenarios. The simulations were performed in Simulink. Eventually the algorithm will be implemented on real hardware, a hardware implementation should be able to process the samples real-time at the rates mentioned in chapter 6. To facilitate implementation of the extended CMA algorithm in hardware a short analysis of its computational complexity is given in section 7.1.

Quantization and operations with finite precision affect the performance of algorithms. Quantization is discussed in section 7.2 and a short look at the fixed-point performance of CMA is given in section 7.3.

7.1 Computational complexity of extended CMA

A cost analysis of the extended CMA algorithm requires the update operation to be split up in basic arithmetic operations that are supported by the hardware platform. First the steering vector update operation is analysed, thereafter a complexity analysis on a system level is discussed.

Update operation complexity

A block diagram of the steering vector update operation can be used to enable clear identification of the different operations. The steering vector update operation has first to be written in causal form before it can be converted into a block diagram. If the update operation is first divided by 4μ the number of multiplications can be decreased. The causal form of the steering vector is derived as follows:

$$\begin{aligned}
\phi[n+1] &= \phi[n] - \mu \cdot \tfrac{8j(|y[k]|^4 - |y[k]|^2) + 4\sin(4y_\varphi[k])}{4j \cdot y[k]} \cdot \boldsymbol{x}[k] \\
\phi[n+1] &= \phi[n] - \tfrac{2j(|y[k]|^4 - |y[k]|^2) + \sin(4y_\varphi[k])}{\frac{j}{\mu} \cdot y[k]} \cdot \boldsymbol{x}[k] \\
\phi[n] &= \phi[n-1] - \tfrac{2j(|y[k-1]|^4 - |y[k-1]|^2) + \sin(4y_\varphi[k-1])}{\frac{j}{\mu} \cdot y[k-1]} \cdot \boldsymbol{x}[k-1]
\end{aligned} \tag{7.1}$$

Figure 7.1 shows the block diagram of the causal update operation.
Most hardware platforms only support real arithmetic operations, therefore the operations from the block diagram are written in terms of real additions, multiplications and divisions. In the analysis of the block diagram operations the number of additions includes the number of subtractions, because the

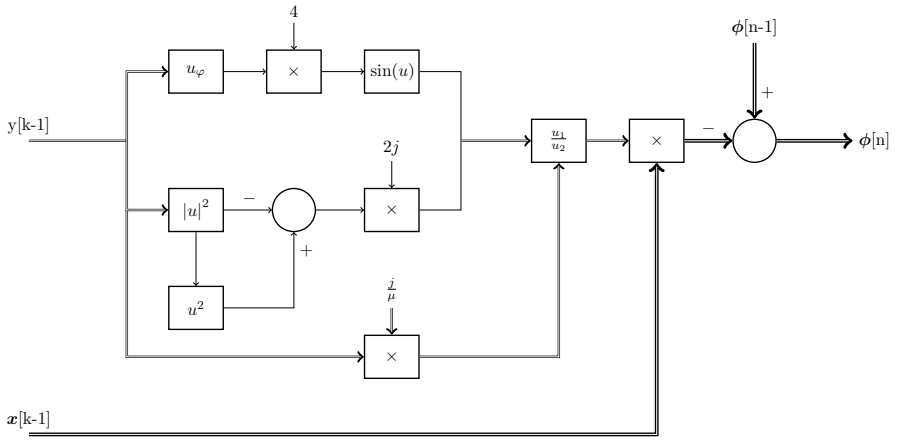

Figure 7.1: Block diagram of the extended CMA algorithm.

addition and subtraction operation use the same hardware if the common two's complement number representation is used [Hey04].

Instantaneous phase determination

Determination of the instantaneous phase of a complex signal can be seen in equation 5.8, phase determination requires division and calculation of the arctangent. Both these operations do not map well into hardware. However, the Coordinate Rotation Digital Computer (CORDIC) algorithm can be used to iteratively perform the required Cartesian to Polar transformation. CORDIC only uses bit shifts and additions to operate [And98]. Generally the iterative form of CORDIC gains one bit of accuracy for each iteration. A full description and complexity analysis of CORDIC is out of scope of this work. A complete survey of the CORDIC algorithm and its wide range of functions can be found in [And98].

Sine function

Three common methods exist to approximate sine functions:

- CORDIC algorithm
- Lookup Table (LUT)
- Taylor series

CORDIC can be used to approximate the sine function using only shift and add operations.

The fastest but most memory inefficient method for approximating the sine function is to use a LUT with precomputed sine values. A LUT only contains approximations for a fixed number of input values, consequently its precision depends on the size of the LUT.

Another method to find approximations for trigonometic functions is calculation of its Taylor series. Precision and complexity is determined by the number

of terms used to approximate the sine function. The Taylor series expansion for the sine function is written as follows [Ste02]:

$$\sin(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1} \tag{7.2}$$

Squared modulus calculation

Calculation of the squared modulus ($|y[k-1]|^2$) can be done by complex multiplication of $y[k]$ by its complex conjugate $y^*[k]$ (property $|z|^2 = zz^*$). In section 2.2 it is shown that complex multiplications can be implemented using four real multiplications and two real additions, but that other methods exist. Conjugation in hardware is no more costly than performing a negation instruction on the imaginary part of the complex number to be conjugated.

Multiplication by a power of two

Two of the four multiplications from figure 7.1 are multiplications by a power of two. Those multiplications can be implemented by bit shift operations. For example multiplication of a binary value with a factor four can be performed by two bit shifts to the left. Depending on the number representation appropriate actions involving the sign bit need to be taken.

Complex division

Complex division is the most costly operation from figure 7.1 to implement. An example of complex division for the quotient $\frac{a+bj}{c+dj}$ is given by:

$$\frac{a+bj}{c+dj} = \frac{a+bj}{c+dj} \cdot \frac{(c+dj)^*}{(c+dj)^*} = \frac{a+bj}{c+dj} \cdot \frac{c-dj}{c-dj} = \frac{ac+bd}{c^2+d^2} + \frac{bc-ad}{c^2+d^2} j \tag{7.3}$$

Both the numerator and denominator of the quotient are multiplied by the complex conjugate of the denominator. This method of complex division requires six real multiplications and two real divisions.

In the complex plane, complex division is rotation of the position vector that represents the complex number and division of its magnitude by the magnitude of the complex denominator of the quotient. During division the direction of rotation is opposite to the direction of rotation in the case of complex multiplication.

CORDIC can be used to efficiently implement complex division based on the geometric meaning of complex numbers. A CORDIC based complex division needs two CORDIC operations, one real division and two real multiplications. An introduction to complex arithmetic using CORDIC can be found in [Hit75].

System level complexity

The computational complexity on a system level is based on the number of times that a certain basic operation is performed. For the extended CMA algorithm this basic operation is the steering vector update. An expression for the computational complexity of the extended CMA algorithm can be found by counting the required operations for a steering vector update.

If CORDIC is performed for the instantaneous phase calculation u_φ and the modulus result of this CORDIC operation is used for the squared modulus calculation then the following expression for the computational complexity can be found:

$$T \cdot (\text{ADD} + 2 \cdot \text{MULT} + 3 \cdot \text{CORDIC} + 2 \cdot \text{BITSHIFT} + (N+1) \cdot \text{CMULT}) \quad (7.4)$$

Herein, T represents the number of simultaneously tracked sources and N is the number of antenna elements. Based on this expression it can be concluded that the computational complexity of the extended CMA algorithm grows linear with T and N.

7.2 Quantization errors

An implementation of the blind beamforming algorithms on digital hardware requires some important design choices regarding quantization. Obviously digital signals are represented by bits, but how many bits are necessary to represent the data values with enough precision? Quantization is a nonlinear and noninvertible process that is hard to analyse analytically. The error due to quantization of a signal is often expressed as additive noise. Obviously, if more bits are used to quantize a signal this additive noise will be smaller. An often used approximation for the relation between the number of bits b and the additive quantization noise is given as follows:

$$\text{SQNR}(dB) = 6.02b + 1.76 \quad (7.5)$$

A derivation of this so-called Signal-to-Quantization-Noise ratio (SQNR) can be found in [PM96]. In chapter 6 it is explained that the minimum required SNR for QPSK demodulation in a DVB-S receiver is 13 dB. If the noise in the complete system would only consist of quantization noise then two bits would suffice to guarantee a 13 dB SNR and thus QEF demodulation. However, the effective number of bits for a real world Analog-to-Digital Converter (ADC) is somewhat lower than its theoretic approximation [PM96]. Therefore, often bigger word sizes are chosen in most embedded hardware designs.

7.3 Finite precision effects

Finite precision affects the performance of the adaptive algorithms. The Matlab fixed-point toolbox is used to execute the CMA algorithm with finite precision operations. One sinus period of the sinusoidal steering angle is simulated with finite precision, the parameters of this simulation can be found in table 6.1.

Two different word lengths are used during this fixed-point simulation. The first simulation uses 16-bit signed words with a 14-bit fraction length. The second simulation uses 24-bit signed words with a 22-bit fraction length. Additions and multiplications are performed in the 'KeepMSB' mode, an explanation of this fixed-point mode can be found in the Matlab documentation [Mat09].
The sinusoidal steering angle simulation for a 16-bit CMA minimizer stopped after a certain time period because of bit errors. The same simulation with a 24-bit minimizer performed flawlessly. The CMA costs for the 16-bit and 24-bit minimizer during the simulation can be seen in figure 7.2. The 16-bit minimizer

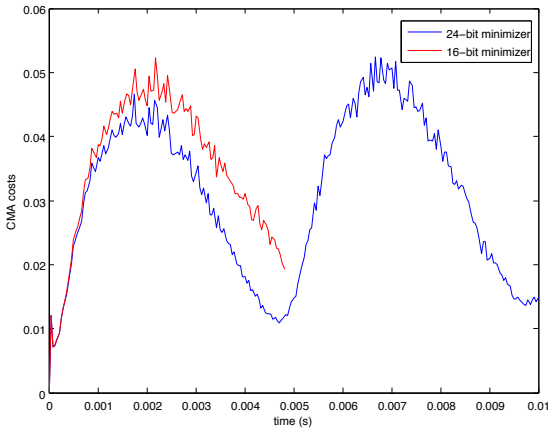

Figure 7.2: Finite precision affects CMA execution.

costs did not increase at the moment of failure, it is not clear why the 16-bit minimizer causes bit errors. Probably due to accumulated precision errors. Future work should investigate the effects of execution with finite precision of the CMA and the extended CMA algorithm.

7.4 Conclusion

The extended CMA algorithm is divided in smaller operations. The possible implementations of these operations and their individual complexities are discussed. Thereafter, an expression is found for the computational complexity for the complete extended CMA algorithm. Based on this expression it can be concluded that the computational complexity of extended CMA grows linear with the number of simultaneously tracked sources and the number of antenna elements.

Fixed-point analysis of the extended CMA algorithm requires a lot of implementation choices to be made before it can be performed. Therefore, only the performance of executing CMA with finite precision operations is shortly discussed. Future work should investigate the effects of finite precision execution on CMA and extended CMA more thoroughly.

8

Discussion

8.1 Exploiting orthogonality

The extended CMA algorithm is used to correct both mispointing and Doppler effects. Mispointing leads to a modulus decrease and Doppler results in rotation of the QPSK constellation points. Those effects are orthogonal to each other. By comparing the block diagram of the original CMA algorithm (figure 7.1) and the diagram of the extended CMA algorithm (figure 5.3) one can easily conclude that the original CMA algorithm has much lower complexity. Therefore, if the CMA algorithm is used to prevent mispointing and another algorithm is used to cope with Doppler effects, could that be less complex than using the extended CMA algorithm and deal with both effects simultaneously? Answering this question requires a complexity analysis of algorithms that only deal with constellation rotations, those algorithms are called derotators in literature [YLC04].

8.2 CMA algorithm rewritten

During this research the idea came up of the possibility to mathematically rewrite the original CMA algorithm in such a manner that its result is not a steering vector but a DOA angle. This would be possible by rewriting the CMA cost function using the DOA θ as the independent variable. Derivation of this cost function with respect to θ can be used to create a minimizer that returns the $\theta[n+1]$ that leads to lower costs. Mathematically this idea (suggested cost function and minimizer) can be written as follows:

$$\begin{aligned} J(\theta) &= E(\left|\boldsymbol{\phi}(\theta)^H \boldsymbol{x}[k]\right|^2 - 1) = E(\left| \begin{bmatrix} 1 \\ e^{2\pi(\frac{d\cdot\sin(\theta)}{\lambda})\cdot 1} \\ \vdots \\ e^{2\pi(\frac{d\cdot\sin(\theta)}{\lambda})\cdot(N-1)} \end{bmatrix}^H \boldsymbol{x}[k] \right|^2 - 1)^2 \\ \theta[n+1] &= \theta[n] - \mu\nabla_\theta J \end{aligned} \tag{8.1}$$

Further work should look into the convergence behaviour of the proposed algorithm.

8.3 Convergence guarantees

The simulations of CMA and extended CMA are only performed for small time periods. Update frequency requirements from section 6.3 are derived from these simulations. During small time periods the algorithms converge correctly for a certain update frequency, however this does not guarantee convergence for longer time periods. Before implementation of the extended CMA algorithm in hardware the chosen update frequency should be thoroughly tested to guarantee convergence.

8.4 Interferer suppression

Simulations of CMA and extended CMA have also shown that these algorithms are well capable of suppressing moving interferers. Due to time constraints these results are not mentioned in this report. Further work should definitely investigate the interferer suppression capabilities of the adaptive algorithms.

8.5 Two-dimensional adaptive arrays

Section 2.1 shortly mentions the existence of beamforming techniques for two-dimensional arrays. Further work should look at the usability of the adaptive algorithms mentioned in this work for two-dimensional arrays.

9

Conclusions

Mobile reception of DVB-S signals requires tracking of the desired signals. Adaptive phased array techniques are used to accomplish this goal, but this requires altering of the original DVB-S receiver chain. The changes consist of adding a beamformer and an adaptive steerer, the proposed system can be seen in figure 9.1. The parts of the original chain are in the DVB-S demodulation subsystem. The newly proposed chain enables DVB-S signal tracking on mobile phased arrays.

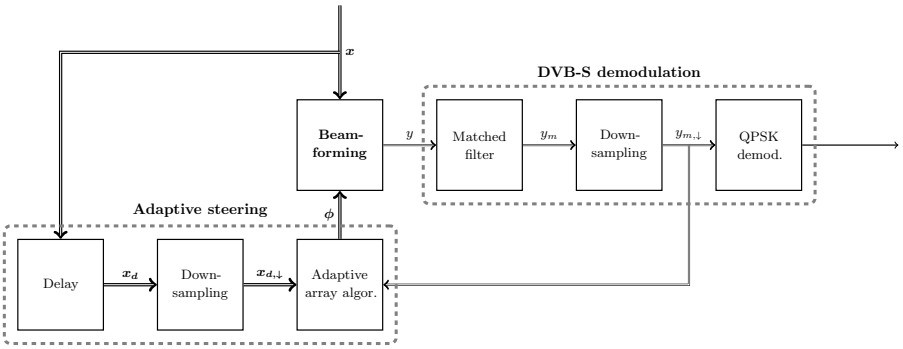

Figure 9.1: Blind beamforming of DVB-S signals.

The beamformer of the proposed system uses phase shift based beamforming. The actual phase shifts are implemented by complex multiplication. A steering vector $\boldsymbol{\phi}$ contains the complex values that represent the required phase compensations to control the directivity of the phased array.

The complex weight values of the steering vector $\boldsymbol{\phi}$ have to dynamically adapt to changing signal conditions. This is done by the adaptive steering subsystem of figure 9.1. There is no reference signal in a DVB-S signal. Therefore, the adaptive algorithm uses structural properties of the signal to perform weight adjustment. Such an adaptive algorithm belongs to the class of blind beamforming algorithms.

The structural property of interest for a DVB-S signal is its QPSK channel modulation. QPSK uses four different phases to represent the transmitted symbols. Those symbols are often drawn in a constellation diagram that expresses assignment of bit patterns to specific output values. Movement of the phased array affects the output of the beamformer. If the phase reference of the array is centered then translational movement of the array leads to rotation

of the QPSK constellation. Rotational movement of the array leads to the orthogonal effect, i.e. a modulus decrease of the QPSK symbols.

Two blind beamforming algorithms are discussed to adjust the steering vector weights: the CMA algorithm and the extended CMA algorithm. Both algorithms define a cost function that is minimized using a gradient descent. The CMA cost function is based on deviations of the beamformer output from a constant modulus. The extended CMA cost function also includes phase deviations of the beamformer output from the equally distributed QPSK phases. The extended CMA algorithm is preferred for steering array sensitivity during vehicle motion because it can simultaneously correct phase and modulus deviations caused by translational and rotational movement of the array.

For simulation of the proposed adaptive DVB-S receiver vehicle dynamics are modelled to generate antenna data that contains the effects of rotational and translational movement of the vehicle. Two vehicle models are discussed: the planar bicycle model and the half-car suspension model. The planar bicycle model is used to model yawing and the half-car suspension model is used to model pitching. Eventually, only the dynamics of the planar bicycle model during one specific scenario were used for antenna data generation. This scenario is called the 'sudden change in steering angle' and considers the translational and rotational dynamics of an Renault during an instantaneous steering angle (11.5 degrees) while driving with a constant velocity of 72 km/h.

Execution of the extended CMA algorithm for antenna data generated by the 'sudden change in steering angle' scenario has shown that this adaptive algorithm is able to flawlessy track the DVB-S signal during the scenario.

The complexity of extended CMA is analysed and an expression for its computational complexity is given. It can be concluded that the computational complexity of this algorithm grows linear with the number of simultaneously tracked sources and the number of antenna elements. Furthermore, effects of finite precision on the original CMA algorithm are mentioned. It is shown for a synthetic simulation scenario that 24-bits words and operations are necessary to function properly. However, further research should look if that number of bits can be reduced and how finite precision operations affect the extended CMA algorithm.

Timing requirements for a hardware implementation are found by downsampling the inputs of the adaptive array algorithm and introducing a ratio that expresses the beamformer sample rate over the adaptive algorithm update rate. The upper bound for this ratio is found by simulation and is in the order of hundreds.

This work will conclude with concise answers to the main questions of this research that are mentioned in the introduction:

- Integration of adaptive phased arrays techniques in the DVB-S chain requires addition of a beamformer and an adaptive steerer to the original system. This adaptive steerer should execute a blind beamforming algorithm because there is no reference signal in a DVB-S broadcast.

- The extended CMA algorithm is the preferred blind beamforming algorithm for execution in the adaptive steerer during vehicle movement. This is due to the fact that it can simultaneously correct phase and modulus deviations caused by translational and rotational movement of the array.

- The computational complexity of extended CMA grows linear with the number of simultaneously tracked sources and the number of antenna elements. Simulations have shown that the extended CMA algorithm is able to update the steering vector correctly in situations where the beamformer sample rate over the steering vector update rate is more than five hundred times larger. These computational complexity and timing requirements can be used for derivation of embedded hardware requirements.

List of Symbols

α_f	Front tire sideslip angle.
α_p	Body pitch angle.
α_r	Rear tire sideslip angle.
β	Raised cosine filter roll-off factor.
β_f	Angle between the longitudinal axis and the front tire velocity vector.
$\boldsymbol{\phi}$	Array steering vector.
$\boldsymbol{A}$	Matrix capturing effects of source and array positions.
$\boldsymbol{E}_N$	Noise subspace.
$\boldsymbol{n}(t)$	Signal and instrumentation noise vector.
$\boldsymbol{R}_{xx}$	Cross-correlation where the input vectors are the same (aka autocorrelation).
$\boldsymbol{S}$	Autocorrelation matrix.
$\boldsymbol{s}(t)$	Vector of signal sources.
$\boldsymbol{v}_f$	Front tire velocity vector.
$\boldsymbol{v}_r$	Rear tire velocity vector.
$\boldsymbol{x}$	Antenna snapshot, the beamformer input.
$\boldsymbol{x}(t)$	Complex antenna snapshot at time t.
$\boldsymbol{x}[k]$	Complex discrete antenna snapshot.
δ	Steering angle.
$\dot{\varphi}$	Phase shift time derivative.
λ	Wavelength of the electromagnetic signal received by the array.
τ_n	Time delay of signal experienced at the n^{th} antenna element.
θ	DOA angle.
$\theta_{\alpha_p=0}$	DOA angle for an array mounted to a vehicle with zero pitch.

B	Minimum required bandwidth.
$C_{\alpha,f}$	Front tire cornering stiffness.
$C_{\alpha,r}$	Rear tire cornering stiffness.
C_{sp}	Suspension damping coefficient.
d	Distance between adjacent antenne elements.
f_h	Highest frequency component of signal band.
f_l	Smallest frequency component of signal band.
f_s	Sampling frequency.
$F_{y,f}$	Front tire lateral force.
$F_{y,r}$	Rear tire lateral force.
I_p	Inertia of half the car body.
I_y	Vehicle yaw inertia.
k	Number of sources in data model.
K_{tf}	Front tire stiffness.
K_{tr}	Rear tire stiffness.
M	Vehicle mass.
M_h	Half the vehicle mass.
M_{usf}	Mass of front wheel and suspension (undamped mass front).
M_{usr}	Mass of rear wheel and suspension (undamped mass rear).
N	Number of adjacent antenna elements.
p	Propagation speed of electromagnic waveform.
P_a	Normalised (logarithmic) power pattern of the array.
P_n	Noise power.
P_s	Signal power.
P_{MU}	MUSIC power spectrum.
r	Vehicle yaw rate.
R_s	Symbol rate.
S_a	The total radiation pattern of the array.
S_n	Complex signal received by the n^{th} antenna element of the ULA.
T	Number of sources simultaneously tracked sources.

T	Symbol time.
$v_{x,CG}$	Longitudinal velocity of vehicle center of gravity.
$v_{y,CG}$	Lateral velocity of vehicle center of gravity.
x_t	Longitudinal axis of the front tire.
y	Beamformer output.
y_m	Matched filtered beamformer output.
$y_{m,\downarrow}$	Downsampled matched filtered beamformer output.

List of Acronyms

DVB-S	Digital Video Broadcasting Satellite
ULA	Uniform Linear Array
FB	Fractional Bandwidth
DOA	Direction of Arrival
FIR	Finite Impulse Response
FFT	Fast Fourier Transform
MUSIC	Multiple Signal Classification
AWGN	Additive White Gaussian Noise
CMA	Constant Modulus Algorithm
QPSK	Quadrature Phase-Shift Keying
SNR	Signal-to-Noise Ratio
CORDIC	Coordinate Rotation Digital Computer
I	In-phase
Q	Quadrature-phase
ISI	Intersymbol Interference
RRC	Root-raised Cosine
QEF	Quasi Error Free
BER	Bit Error Rate
LUT	Lookup Table
SQNR	Signal-to-Quantization-Noise ratio
ADC	Analog-to-Digital Converter
SER	Symbol Error Rate

Bibliography

[AG05] B. Allen and M. Ghavami. *Adaptive Array Systems, Fundamentals and Applications*. John Wiley Sons, 2005.

[And98] R. Andraka. A survey of CORDIC algorithms for FPGA based computers. *FPGA 98*, 1998.

[Com08] D. Comer. *Computer Networks and Internets*. Prentice Hall, 2008.

[ETS97] Digital video broadcasting (DVB) framing structure, channel coding and modulation for 11/12 GHz satellite services, 1997.

[G3D09] Google 3D warehouse object. Google 3D Warehouse, 2009.

[Gen02] K. Gentile. The care and feeding of digital, pulse-shaping filters. *RF Design Magazine*, 2002.

[God04] Lal Chand Godara. *Smart Antennas*. CRC Press, 2004.

[Gom02] Jorge Matos Gomez. *Satellite Broadcast Systems Engineering*. Artech House Publishers, 2002.

[Hay98] Toby Haynes. A primer on digital beamforming. Spectrum Signal Processing White Paper, 1998.

[Hey04] P.M. Heysters. *Coarse-Grained Reconfigurable Processors*. PhD thesis, University of Twente, 2004.

[HH89] P. Horowitz and W. Hill. *The art of electronics*. Cambridge University Press, 2nd edition, 1989.

[Hit75] S. Hitotumatu. Complex arithmetic through CORDIC. *Kodai Math. Sem. Rep.*, 1975.

[Jaz08] Reza N. Jazar. *Vehicle Dynamics: Theory and Application*. Springer, 2008.

[KHJ05] R. Khosla, R. Howlett, and L. Jain. Knowledge-based intelligent information and engineering systems. 2005.

[Lyo99] Rick Lyons. The discrete hilbert transform: A brief tutorial. In *International Conference on Signal Processing Applications and Technology*, 1999.

[Lyo04] Rick Lyons. *Understanding Digital Signal Processing*. Prentice Hall, 2004.

[Mat09] The Mathworks. Matlab, 2009. R2009a.

[MM04] Robert A. Monzingo and Thomas W. Miller. *Introduction to Adaptive Arrays*. SciTech Publishing, 2004.

[PBA03] Timothy Pratt, Charles Bostian, and Jeremy Allnut. *Satellite Communication*. Wiley, 2003.

[PM96] John G. Proakis and Dimitris K Manolakis. *Digital Signal Processing: Principles, Algorithms, and Applications*. Prentice Hall, 1996.

[Pro01] J. Proakis. *Digital Communications*. McGraw-Hill, 2001.

[RSF01] G. Rideout, J. Stein, and B. Ferris. Target cascading: A design process for achieving vehicle ride and handling targets. In *Proceedings of IMECE*, 2001.

[Rud82] A.W. Rudge. *The Handbook of Antenna Design: Volume 1*. Institution of Electrical Engineers, 1982.

[Sch86] R. Schmidt. Multiple emitter location and signal parameter estimation. *Antennas and Propagation, IEEE Transactions on*, 34(3):276–280, Mar 1986.

[SGF07] J. Sergio, D. Gonzalo, and R. Facundo. On commutation modeling in bond graphs. 2007.

[Sko01] Merril I. Skolnik. *Introduction to Radar Systems*. McGraw Hill, 2001.

[Smi07] Julius O. Smith. *Mathematics of the Discrete Fourier Transform (DFT) with Audio Applications*. W3K Publishing, 2007.

[Ste00] Jonathan Y. Stein. *Digital Signal Processing: A Computer Science Perspective*. John Wiley & Sons, 2000.

[Ste02] J. Stewart. *Calculus: Early Transcendentals*. Brooks Cole, 2002.

[TA83] J. Treichler and B. Agee. A new approach to multipath correction of constant modulus signals. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 1983.

[vdV04] Alle-Jan van der Veen. *Signal Processing for Communications*, 2004. Reader ET4 147.

[Vis05] Hubregt J. Visser. *Array and Phased Array Antenna Basics*. John Wiley & Sons, 2005.

[Wer08] Johan Wernehag. Microwave CMOS beamforming transmitters. Master's thesis, Lund University, 2008.

[YC05] C. Yao and C. Chien. The design of a square-root-raised-cosine FIR filter by a recursive method. 2005.

[YLC04] H. Yang, Z. Lin, and X. Cai. Design of a QPSK demodulator for DVB-S receiver ASIC chip. 2004.

[Zhe00] Xu Zhengyuan. New cost function for blind estimation of M-PSK signals. *Wireless Communications and Networking Conference*, 3, 2000.

A

Half-car suspension model

The equations of motion for figure 3.4 are derived in [Jaz08]. The resulting set of equations in matrix form is shown below:

$$
\begin{aligned}
&\boldsymbol{M}\ddot{\boldsymbol{x}} + \boldsymbol{C}\dot{\boldsymbol{x}} + \boldsymbol{K}\boldsymbol{x} = \boldsymbol{f} \\
\boldsymbol{x} &= \begin{bmatrix} x \\ \alpha_p \\ x_1 \\ x_2 \end{bmatrix} \\
\boldsymbol{M} &= \begin{bmatrix} M_h & 0 & 0 & 0 \\ 0 & I_p & 0 & 0 \\ 0 & 0 & M_{usf} & 0 \\ 0 & 0 & 0 & M_{usr} \end{bmatrix} \\
\boldsymbol{C} &= \begin{bmatrix} 2C_{sp} & C_{sp}d_r - C_{sp}d_f & -C_{sp} & -C_{sp} \\ d_rC_{sp} - d_fC_{sp} & C_{sp}d_f^2 + C_{sp}d_r^2 & d_fC_{sp} & -d_rC_{sp} \\ -C_{sp} & d_fC_{sp} & C_{sp} & 0 \\ -C_{sp} & -d_rC_{sp} & 0 & C_{sp} \end{bmatrix} \\
\boldsymbol{K} &= \begin{bmatrix} K_{sf} + K_{sr} & d_rK_{sr} - d_fK_{sf} & -K_{sf} & -K_{sr} \\ d_rK_{sr} - d_fK_{sf} & K_{sf}d_f^2 + K_{sr}d_r^2 & d_fK_{sf} & -d_rK_{sr} \\ -K_{sf} & d_fK_{sf} & K_{sf} + K_{tf} & 0 \\ -K_{sr} & -d_rK_{sr} & 0 & K_{sr} + K_{tr} \end{bmatrix} \\
\boldsymbol{f} &= \begin{bmatrix} 0 \\ 0 \\ y_1K_{tf} \\ y_2K_{tr} \end{bmatrix}
\end{aligned}
\tag{A.1}
$$