IT Licentiate thesis
2001-012

Teleconferencing, System Identification and Array Processing

PER ÅHGREN

UPPSALA UNIVERSITY
Department of Information Technology

UPPSALA UNIVERSITY

Teleconferencing, System Identification and Array Processing

BY

PER ÅHGREN

October 2001

DEPARTMENT OF SYSTEMS AND CONTROL
INFORMATION TECHNOLOGY
UPPSALA UNIVERSITY
UPPSALA
SWEDEN

Dissertation for the degree of Licentiate of Philosophy in Signal Processing
at Uppsala University 2001

Teleconferencing, System Identification and Array Processing

Per Åhgren
Per.Ahgren@syscon.uu.se

Department of Systems and Control
Information Technology
Uppsala University
Box 337
SE-751 05 Uppsala
Sweden

http://www.it.uu.se/

ISSN 1404-5117
Printed by the University Printers, Ekonomikum, Uppsala, Sweden

Abstract

The area of teleconferencing has long yielded great interest in the signal processing community. The main reasons for this are probably the huge interest from the industry and the challenging problems of the topic. The problems of teleconferencing are relevant for several different disciplines in signal processing. Three of these are Acoustic Echo Cancellation, System Identification and Sensor Array Signal Processing. In this thesis some problems related to these disciplines are studied. The thesis is divided into 6 parts, one for each paper included.

In the first part a new adaptive algorithm is applied to the acoustic echo cancellation problem. It is shown to perform much better than the Normalized Least Mean Squares (NLMS) algorithm and while it performs worse than the standard Recursive Least Squares (RLS) algorithm it is shown to be computationally simpler than this.

In the second part the hierarchical RLS algorithm is analyzed. The extraordinary results presented for this algorithm in previous papers are discussed and explained.

In the third part a new initialization method for RLS is presented that yields the exact Least Squares estimates while not being computationally more demanding than RLS. This is an important contribution since the standard initialization of the RLS algorithm is somewhat arbitrary.

In the fourth part a method is presented that deals with the problem of estimating the common factors out of an arbitrary number of polynomials. Two problems of array processing and system identification are stated as problems for common factor estimation and the presented method is applied to these. For these two problems the method is shown to perform better than existing methods.

In the fifth part a method for beamforming using few sensors is presented. Data-dependent beamformers usually perform badly when there are few sensors in the array, particularly when the beamformer constraints are numerous. The method presented deals with this problem by approximately fulfilling the beamformer constraints and hence getting extra degrees of freedom for suppressing interferences.

In the sixth part the previously unsolved problem of array processing of non-zero mean signals is solved for the colored noise case. Methods are presented both for the estimation problem and the detection problem and are shown to perform well in numerical examples.

Keywords

Teleconferencing, Acoustic Echo Cancellation, System Identification, Sensor Array Signal Processing, Beamforming.

Acknowledgements[1]

First of all I would like to express my gratitude to my supervisor, Professor Petre Stoica for his invaluable guidance in the world of research. Without him this thesis would never have come into being. I would also like to give special thanks to Doctor Monika Agrawal for her great help and support in our work together, and also to Erik G. Larsson for his valuable advice when developing the results for our joint paper. Then I would like to thank all my other colleagues and the staff at the Department of Systems and Control for making working there so fun and pleasant. I would also like to thank Docent Sven Nordebo from the Department of Telecommunications and Signal Processing at Blekinge Institute of Technology for accepting to be my opponent. Finally I would like to thank my family and my girlfriend for putting up with me during these years of research.

[1]This work was partly supported by the Swedish Foundation for Strategic Research (SSF).

Contents

Chapter 1

Introduction

This thesis is about the two major areas in signal processing that I have been involved in during my time as a PhD student. The first area is Acoustic Echo Cancellation (AEC) and the other is Sensor Array Processing (SAP).

1.1 Acoustic Echo Cancellation

Echo cancellers have been used for quite a long time. In the early days of telephony the need first arose since electrical echoes were generated in the splices between different telephone line segments. These echoes made it practically impossible to use the telephones since they were propagated back and forth between the different splices with very little attenuation. To solve this problem line echo cancellers [7][37] were constructed that removed the echoes by first predicting the echoes generated from a certain signal and then removing the predicted echo from the received signal by simple subtraction (note that we are dealing with electrical signals so subtraction is done in a simple manner). These echo cancellers were very successful and today, thanks to these, almost no echo at all can be perceived when using telephones.

In more recent years the need for echo cancellation has appeared in a new area, namely hands-free telephony [12] for which the demand is expected to increase enormously in the years to come. In the beginning of telephony the users had to use both their hands to hold the loudspeaker and the microphone [38]. Soon the microphone and loudspeaker were combined into a handset which use required one hand only. The next step is called hands-free telephony in which the user have both hands free. In hands-free telephony acoustic echoes are generated by the acoustic feed-back loop in the room where the telephone is located. This means that the sound emitted by the loudspeaker is picked up by the microphone and sent back as an echo. Thus a speaker in the other end will hear an echo of his own voice when he speaks to the person with the hands-free telephone setup. If the delay in the echo is small, it will not be noticed. This is the case when we talk in a room and are not disturbed by the echoes of our own voice. In telephony, however, there is the additional delay for the signals to travel via the telephone network. This can be quite long, depending on the telephone setup and connection, and the echoes may be noticeable. Therefore, in some way the echoes from the loudspeaker output that are picked up by the microphone must be removed from the microphone signal. That is what acoustic echo cancellation [5] is about. Note that if the attenuation between the loudspeaker and the microphone is high, the echoes will be negligible, which is why we do not need acoustic echo cancellation for traditional telephone setups.

1.1.1 Practical applications for acoustic echo cancellation

Some of the applications for AEC are:

- Hands-free telephony.
- Teleconferencing systems.
- Internet telephony.
- Hearing aids.
- Public addressing systems.

All of these can in theory be solved by the techniques discussed here. For some of them, however, the requirements are very rigorous and for these, in practice, the techniques discussed here are not usable. In hearing aids, for instance, the computational requirements are extremely strict due to the limited battery power of hearing aids. For public addressing systems the rooms involved are so large that the computational requirements for applying the techniques discussed here are enormous.

1.1.2 Loudspeaker enclosure microphone systems

In AEC the central components are a loudspeaker and a microphone placed within one enclosure. This system is called a Loudspeaker-Enclosure-Microphone (LEM) system [5] and a simple model of such a system is depicted in Figure 1.1. If there is no overload of the A/D converters this system may

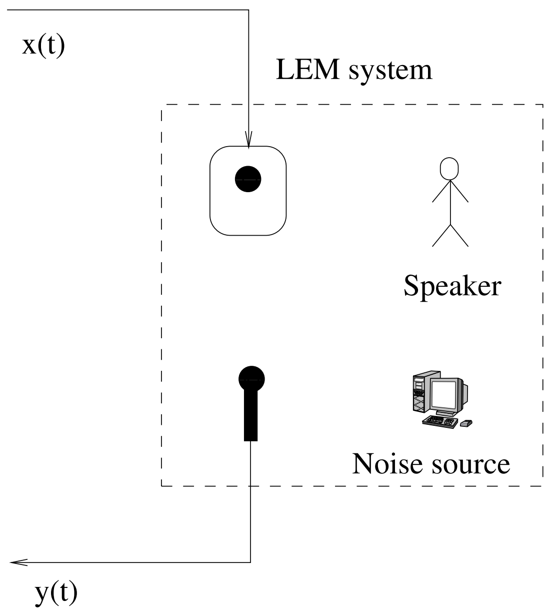

Figure 1.1: A Loudspeaker-Enclosure-Microphone (LEM) system.

be modeled with sufficient accuracy as a linear system. The impulse response of the system is dependent upon the different properties of the enclosure [1], the material in the furniture, the location of different objects, etc., and is highly sensitive to modifications in these [5][8]. Since generally there are objects moving in the room this means that the impulse response of the rooms is highly time-varying.

For some systems this dependence of the acoustic path on the properties of the enclosure makes it possible to successfully remove the echoes by modifying the acoustic properties of the enclosure. For instance we can cover the walls in a room with sound-dampening plates and adjust the placement of the loudspeaker and microphone to reduce the echoes by reducing the reverberation time of the room. But for many systems this is not possible and therefore we have reduce the echoes by electronical means.

Figure 1.2 shows a typical impulse response of a LEM system, sampled in 16 kHz. The impulse response can be seen as the resulting microphone output

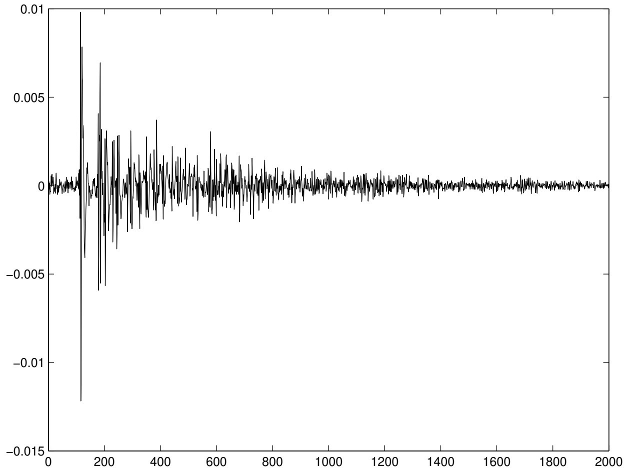

Figure 1.2: Impulse response of a typical LEM system (16 kHz sampling frequency).

signal when a very short burst of sound is send to the loudspeaker. As is easily seen the impulse response is almost zero until about sampling instant 100. This corresponds to the direct path for the sound propagation between the loudspeaker and the microphone. The rest of the impulse response consists of superimposed echoes corresponding to all the other possible echo paths with amplitudes depending on the lengths of the echo paths and the properties of the materials the sound waves have been reflected from. The reverberation time of a LEM system is generally of the order of a few tens to a few hundreds of milliseconds depending on the properties of the enclosure [8].

In Figure 1.1 a typical LEM system for the near-end (the end of the communication line where the echo cancellation is to take place) is depicted with the loudspeaker and the microphone. A near-end speaker and a noise source (a computer) are also present. The signal to the loudspeaker (sent from the far-end room (the other end of the communication line)) is denoted by $x(t)$ and the signal from the microphone (sent to the far-end room) is denoted by $y(t)$. This LEM system is usually modeled as a linear system. In Figure 1.3 a linear model of the LEM system in Figure 1.1 is shown. In this

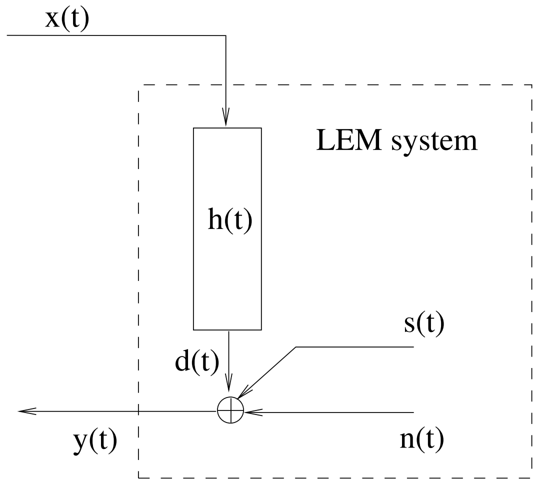

Figure 1.3: Linear model of a LEM system.

the transmission path of the sound signal is modeled as a linear system with impulse response $h(t)$ consisting of the complete path from where the signal $x(t)$ enters the LEM system to where it exits the LEM system as the signal $y(t)$. Thus, the responses of the microphone and the loudspeaker are included in $h(t)$ as well as the echo path within the room. The near-end speaker sound signal $s(t)$ and the near-end noise source signal $n(t)$ are simply added to the microphone signal (note that a thing not shown in the figure is that these will also be affected by the response of the microphone).

Industrial requirements

The characteristics of the telephony (in the mono case) are regulated by the International Telecommunication Union (ITU). Their most critical demands are the allowable delay times in the telephone circuitry, 2 ms for stationary telephones and 39 ms for mobile phones (the higher allowable delay time for mobile phones is what makes the echo more severe when mobile phones are involved) [8][5]. In an ordinary telephone the handset setup gives an attenuation in the LEM system of at least 46 dB and this is also required for a hands-free telephone by the ITU. Early hands-free telephony setups achieved this attenuation by allowing only half-duplex communication, that is, communication in only one direction at a time. However, full-duplex communication (communication in both directions at the same time) is desirable. Then AEC is required to achieve the desired attenuation. Achieving an attenuation of 46 dB using an echo canceller is, however, a very hard task but fortunately, the attenuation of 46 dB is only required during single-talk (when there is speech generated only in one end of the communication line). During double-talk (when speech is generated in both ends of the communications line) only an attenuation of 29 dB is required [8]. Therefore, as we will see, the echo canceller is in practice combined with other functionality in order to achieve this goal.

A typical setup for an acoustic echo canceller

A typical setup for an acoustic echo canceller (mono) is depicted in Figure 1.4. As in Figure 1.3 the LEM system transfer function is denoted by $h(t)$ and by

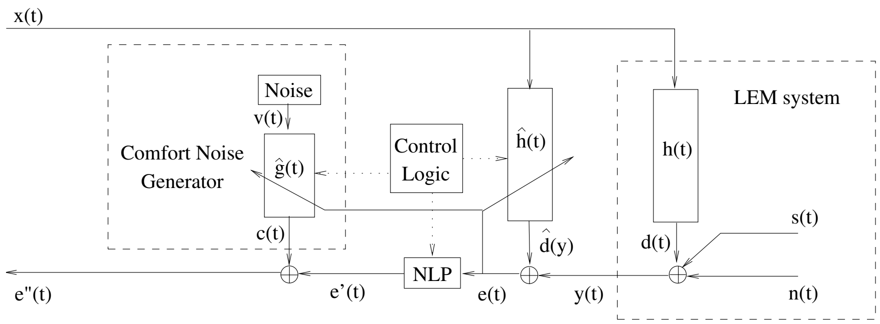

Figure 1.4: A general setup for acoustic echo cancellation for a LEM system.

$\hat{h}(t)$ we denote an estimate of the LEM system transfer function. The input signal to the LEM system and to the estimate of the LEM system is denoted by $x(t)$, and the LEM system output is denoted by $y(t)$. The output of the estimated LEM system is denoted by $\hat{d}(t)$. As is seen in the figure speech and noise are generated in the LEM system as the signals $s(t)$ and $n(t)$, and we have that the LEM system output is given by

$$y(t) = d(t) + s(t) + n(t), \tag{1.1}$$

where $d(t)$ is the contribution to the LEM system output by the input to the LEM system. The echo cancellation is performed by subtracting the LEM system estimate output $\hat{d}(t)$ from the microphone output signal $y(t)$ giving the resulting difference $e(t)$. Ideally we would like that $\hat{d}(t) = d(t), \forall t$. In that case we will have that for the noise-free case

$$e(t) = y(t) - \hat{d}(t) = s(t) \tag{1.2}$$

which means that only the locally generated talk and not the sound from the loudspeaker will be passed through the echo canceller.

As is mentioned above the requirements for echo-suppression during double-talk and single-talk differ. During double-talk it is possible to obtain the required echo suppression by echo cancellation (that reduces the echo by 10-35 dB). Achieving the required echo suppression during single talk is, however, a much harder task. The most common way to achieve the required suppression is therefore to use a Non-Linear Processor (NLP) that is in use only during single-talk. As is shown in Figure 1.4 the NLP is connected to the echo canceller output $e(t)$. During single-talk, when no near-end talk is generated ($s(t) \equiv 0$), the NLP is turned on and since we then know that the output of the echo canceller ideally should be zero the NLP simply sets $e(t) \equiv 0$ and the echo suppression requirements are achieved. Unfortunately this will be clearly audible in the other end of the line since the signal will be completely noise-free, which is not the case normally. To do this we have to add a comfort noise signal $\hat{n}(t)$ that is an estimate of the background noise $n(t)$ in the LEM system.

To control when the NLP should be turned on we have to have some kind of control logic in the acoustic echo canceller. This is also needed since most algorithms for AEC can only update the impulse response estimate during single-talk. The main parts of the control logic are a double-talk detector that detects when there is speech both in the near-end room ($s(t) \neq 0$) and in the loudspeaker signal ($x(t) \neq 0$), and a detector that detects when there is far-end speech.

1.1.3 Adaptive filtering

In the previous section we set up a model for the system in Figure 1.3. This model can be designed in many different ways. What is common for most of these models is, however, that they have some system dependent parameters that describes the properties of the system. Usually the system model can be described in terms of a linear filter (where the system parameters are the filter coefficients), either as an Infinite Impulse Response (IIR) [13] filter, or as a Finite Impulse Response (FIR) [13]. The IIR filter is described by the difference equation

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1+\sum_{k=1}^{N} a_k z^{-k}} x(n)\,, \qquad (1.3)$$

where $\{a_k, b_k\}$ are the IIR system parameters and $y(k)$ and $x(k)$ are the filter output and input signals respectively. The IIR filter is good for filtering narrow frequency peaks in the data and is also good for modeling systems with long impulse responses since it has an infinite impulse response. A big drawback is, however, that it may be unstable if the parameters are chosen such that the the polynomial $1+\sum_{k=1}^{N} a_k z^{-k}$ have zeros with modulus greater than 1.

The FIR filter is a special case of the IIR filter when all $a_k = 0, \forall k$. It is inherently stable (since it has no poles) but has the drawback that the impulse response of the FIR filter is limited in time by the number of filter coefficients. Another drawback is that it is bad for modeling narrow frequency bands since the transfer function has no poles.

In AEC the FIR filter is usually the preferred choice. The reasons for this is its inherent stability but also that a large number of filter parameters are required to successfully model the acoustic impulse response that has a highly detailed and irregular shape. Therefore the IIR filter has no advantages against the FIR filter [15][18].

The values of the filter parameters for a system are usually estimated from measured values of the inputs and outputs of the system. This can be done using a number of different methods. In AEC the LEM system is generally time-varying which means that the filter parameter estimates of the system have to be updated over time. This is done by adaptive filtering in which the values of the filter parameters are updated adaptively while using the filter.

There are a vast number of methods available for adaptive estimation of the filter parameters and here we will only cover those that are of greatest importance for the AEC problem. General for all of them is that they all work with the linear regression model

$$y(t) = \varphi^H(t)\theta + n(t), \qquad t = 1, \ldots, N, \qquad (1.4)$$

where N denotes the number of available data samples: $\{y(t), \varphi(t)\}_{t=1}^{N}$, $n(t)$ is a noise term, θ is the vector of unknown parameters with $\dim(\theta) \triangleq n$, and the superscript "$H$" stands for the conjugate transpose (or just the transpose for real-valued data). (Note that filter models mentioned above can easily be written on this form).

The performance of the different adaptive algorithms are primarily measured by the following properties:

- *Misadjustment* The misadjustment of the algorithm concerns how much the parameters differ from their true values.

- *Computational requirements* This concerns the number of arithmetic operations required to update each filter parameter. Mostly divisions and multiplications are counted since these take much longer time to perform than additions or subtractions.

- *Convergence rate* This concerns how fast the algorithm will change the filter parameters to their final values.

- *Tracking* Tracking concerns how the algorithm will respond to changes in the true parameters.

- *Robustness* The algorithms should be robust to disturbances in the signals and the model. Small disturbances such as noise should not result in large modifications of the filter parameters.

- *Numerical properties* The algorithms should be numerically stable in the sense that they should not be sensitive to quantization errors. These quantization errors occur when the algorithms are implemented on digital computers that always have a finite word-length.

- *Stability* An algorithm is said to be stable if the mean-squared error converges to a final (finite) value.

Recursive Least Squares

The Least Squares (LS) algorithm [13] is perhaps the most well-known algorithm for filter parameter estimation. The LS estimate $\hat{\theta}$ of the parameter vector θ is given by

$$\hat{\theta}_N = R_N^{-1} r_N, \tag{1.5}$$

where the index N is meant to show that the corresponding quantity was obtained from N data samples, and

$$r_N = \sum_{t=1}^{N} \varphi(t) y(t), \tag{1.6}$$

$$R_N = \sum_{t=1}^{N} \varphi(t) \varphi^H(t). \tag{1.7}$$

$$\tag{1.8}$$

Due to its computational complexity the LS algorithm is, however, not suitable for online estimation (because of the inverse of R_N). Fortunately it can easily be presented in a recursive manner using the matrix inversion lemma [34]. The classical Recursive LS algorithm is called RLS [16] and consists of the following equations:

$$\hat{\theta}_N = \hat{\theta}_{N-1} + K_N e(N), \tag{1.9a}$$

$$e(N) = y(N) - \varphi^H(N)\hat{\theta}_{N-1}, \tag{1.9b}$$

$$K_N = \frac{P_{N-1}\varphi(N)}{\lambda + \varphi^H(N) P_{N-1} \varphi(N)}, \tag{1.9c}$$

$$P_N = \lambda^{-1} P_{N-1} - \lambda^{-1} K_N \varphi^H(N) P_{N-1}, \tag{1.9d}$$

where $P_N = R_N^{-1}$ (this inverse is assumed to exist, hence the RLS algorithm in (1.9) is not applicable for $N < n$), and λ is a forgetting factor. The RLS algorithm must be provided with initial values for $\hat{\theta}_N$ and P_N and the usual way of doing this is to set

$$\hat{\theta}_1 = 0 \qquad \text{and} \qquad P_1 = cI, \tag{1.10}$$

where $c > 0$ is a "suitable chosen" constant. The forgetting factor λ in the RLS algorithm is set to a value between $0 < \lambda \leq 1$ [16], and has the purpose of making RLS adapt to changes in θ by weighting the regressors $\varphi(t)$ with powers of λ, thereby "forgetting" old measurements. The RLS algorithm is easily found to be of computational and storage complexity $\mathcal{O}(n^2)$. There are also sliding window versions of RLS where only the data in a sliding window is considered for computation of the covariance matrix.

Fast transversal filter

A computational simpler version of RLS that seem to be widely used is the Fast Transversal Filter (FTF) [32] algorithm where the special structure of R_N^{-1} has been exploited. It is of complexity $\mathcal{O}(8n)$ which is much lower than that of RLS but, unfortunately, the FTF algorithm suffers from stability problems. In [20] it has been shown that all fast versions of RLS suffer from stability problems when the forgetting factor λ is less than one (which is a requirement for the algorithm to be able to adapt to parameter changes). Therefore control of the propagation of numerical errors has to be included for the algorithm to work well. Another drawback is that these fast versions of RLS can lock (the algorithm converges to a parameter vector not equal to the true one) with speech as input signal (since speech is poorly exciting). This locking can, however, be prevented by conditional reinitialization of the algorithm.

Least Mean Squares

The Least Mean Squares (LMS) [13] algorithm is perhaps the most used algorithm for adaptive filtering applications, primarily due to its simplicity and low implementation cost. The algorithm is defined by the equations

$$e(t) = y(t) - \hat{\theta}^T(t-1)\varphi(t), \tag{1.11a}$$

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \mu\varphi(t)e(t), \tag{1.11b}$$

where μ is an adaptation step size parameter and the other vectors are defined as above. A perhaps more common variant of LMS, called Normalized LMS (NLMS) [13], is defined as

$$e(t) = y(t) - \hat{\theta}^T(t-1)\varphi(t), \tag{1.12a}$$

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{\mu}{a + ||\varphi(t)||^2}\varphi(t)e(t), \tag{1.12b}$$

where a is a small number with the purpose of avoiding division by zero in case $||\phi(t)||^2 = 0$. The stability and convergence properties of (N)LMS are determined by the step-size parameter μ. If μ is too large then the algorithm will not be convergent in a mean square sense. If, on the other hand, μ is too small, then the convergence of the algorithm will be very slow. It can be shown that the LMS algorithm is stable for $0 < \mu < \frac{2}{\lambda_{max}}$ where λ_{max} is the largest eigenvalue of the correlation matrix R of the input data, defined in (1.7). The NLMS algorithm is stable for $0 < \mu < 2$ and the stability of NLMS is thereby independent of the properties of the input signal. Furthermore, NLMS exhibits a rate of convergence that is potentially faster than that of LMS for both correlated and uncorrelated data. The NLMS algorithm is very computationally efficient and is of computational complexity $\mathcal{O}(N)$.

Affine Projection

The affine projection (AP) algorithm [25] can be considered as an extension of the NLMS algorithm, taking into account the last P input vectors instead of just the last one that is the case for NLMS. Its complexity is between that of NLMS and that of RLS. The algorithm is given by:

$$e(t) = y(t) - \hat{\theta}^T(t-1)\varphi(t), \tag{1.13a}$$

$$\bar{e}(t) = [e(t)\, e(t-1)\, \ldots\, e(t-P+1)]^T, \tag{1.13b}$$

$$\Phi(t) = [\varphi(t)\, \varphi(t-1)\, \ldots\, \varphi(t-P+1)]^T, \tag{1.13c}$$

$$\hat{\theta}(t+1) = \hat{\theta}(t) + \mu\Phi(t)(\Phi^T(t)\Phi(t))^{-1}\bar{e}^T(t). \tag{1.13d}$$

where μ is an adaptation step size parameter.

In the AP algorithm [25] the matrix $(\Phi^T(t)\Phi(t))$ has to be inverted. Fortunately this inversion can be carried out recursively and furthermore the dimension P of this matrix is usually relatively small compared to the total number of coefficients to be estimated. Note that if $P = 2$, then the inversion can be calculated directly requiring only a small computational load. Since even a second order AP gives a big improvement over the NLMS algorithm this is very favorable.

A problem with the NLMS algorithm is that the convergence speed is not constant and it has been shown that the convergence speed decreases as the first order autocorrelation function of the signal approaches 1 in absolute value [5]. This problem arises because the direction of coefficient update of the NLMS can be viewed as a one dimensional projection. The AP algorithm overcomes this limitation by cancelling several aposteriori errors in each coefficient update. These coefficient updates may be viewed as multidimensional projections.

A fast version of AP has been developed (FAP) [11] but in this algorithm numerical instabilities occur due to recursively calculated correlation matrices. This problem can, however, be overcome by regularizing the correlation matrix and reinitializing the algorithm whenever divergence occurs.

Sub-band algorithms

The most straightforward way to build adaptive filters is to deal with the full-band signals (the unmodified signals). The most straightforward way however, does not have to be the most efficient. One alternative is to deal with sub-band structures. By using filter banks [13], the signals $x(k)$ and $y(k)$ are split into $M > 1$ frequency sub-bands, each of which an adaptive filter is applied on. Depending on the properties of the bandpass filters [26] the sampling rate in the sub-bands can be reduced and because of this the lengths of the adaptive filters for the different sub-bands can be reduced. Instead of filtering the full-band signal and adapting one single filter at a high sampling rate, M adaptations with sub-sampled signals are performed in parallel at reduced sampling rates. The final output signal $e(k)$ is obtained by recombining and up-sampling the output signals from the different filters.

All the algorithms mentioned above can successfully be applied to the sub-band scheme. NLMS, for instance, has a convergence speed that is depending on the flatness of the input signal spectrum. If the signal is divided into several different sub-bands, each sub-band will have a flatter spectrum than the original signal and the convergence speed will be improved. Another advantage with using NLMS with sub-band structures is that different stepsizes may be used in NLMS for different sub-bands. Another advantage is that the sub-band structure is well suited for parallel implementations.

Sub-band realizations have the substantial disadvantage that they introduce a delay into the system. This delay is caused by the filter banks for signal decomposition and synthesis of the excitation and error signals. Furthermore, when designing the filterbanks the special demands of an adaptive echo canceller has to be taken into consideration. For instance, special care has to be taken to minimize the aliasing terms between different filter banks.

Frequency domain and block adaptive algorithms

The computational complexity for processing long time-domain adaptive filters may be immense due to the long time-domain adaptive filters. Therefore, algorithms with reduced numerical complexity are desired. To decrease the complexity we can use adaptive filters based on block processing. These may work both in the time and frequency domains. The basic principle for block processing algorithm is that a number M of input signals are collected before computing a block of M output signals. Thus the filter is only adapted once every Mth sampling instant and more computing time is available for the filter adaptation. The block processing also makes it possible to use more advanced and efficient signal processing techniques, like fast transforms, to reduce the computational efficiency. Block algorithms are also easily parallelized and pipelined. As for the sub-band processing a big disadvantage to the block processing algoritms is that the block processing introduces a delay into the system.

Adaptive filtering applied to acoustic echo cancellation

Because of the long adaptive filters involved in AEC the computational and storage requirements are hard for the AEC algorithms. The most common method for acoustic echo cancellation is probably NLMS due to its simplicity, and low computational and memory requirements. Better convergence properties are, however, desired and therefore the AP algorithm is also popular, being between NLMS and RLS both in terms of computational complexity and convergence speed. RLS is in practice unusable due to its huge computational complexity and memory requirements. Fast versions of RLS like FTF are, however, used in combination with stability control functionality.

Speech, the input signal in acoustic echo cancellation

The performance of the algorithms for adaptive filtering mentioned above depends heavily on the properties of the input and error signals. In AEC both consists of speech (during double-talk the error signal will contain speech even when the echo cancellation is perfect). Unfortunately, speech is highly non-stationary and can only be considered to be stationary during short time periods. It is widely accepted that parameters derived from a speech signal have to be updated at intervals of about 20 ms. Furthermore, the short-time spectrum for the speech signal is highly colored and thus the covariance matrix for the input signal is ill-conditioned which results in low convergence speed in many algorithms such as NLMS. Even for sampling frequencies of 8 kHz the correlation between neighboring samples in a speech signal obtains values between 0.8 and 0.9. This means that speech signals do not provide a persistently exciting signal for most adaptive algorithms.

The human ear can distinguish frequencies between 20 Hz and 20 kHz. Therefore the sampling frequency should ideally be at least 40 kHz (twice the highest frequency according to the sampling theorem) to avoid aliasing. This would, however, require that the filter output would have to be computed 40000 times a second which is relatively computationally burdensome. Furthermore, if FIR filters were used for echo cancellation (which is usually the case) then the lengths of the FIR filters would be prohibitively large (since the FIR filters have to be long enough to accommodate for the reverberation in the room). The sampling frequencies for AEC applications are for these reasons between 8 and 16 kHz (and this is also the standard in most of the telephone networks of today).

Since speech signals have pauses, it is important to include functionality so that the adaptation of the adaptive algorithms is turned off when there is no speech so that the adaptation does not take place when there is no input signal (or when the input signal is very low). This is easily done by a speech activity detector that usually simply consists of an energy detector that compares the measured energy in the speech signal to a threshold to determine if there is speech present or not. The energy estimate is usually obtained by taking the mean of the squared input values during a given number of samples.

Double-talk detection

There are some algorithms for AEC that allow adaptation during double-talk (are robust to double-talk). For most algorithms, though, some kind of control has to be incorporated so that the adaptation does not take place during double-talk. To this end a double-talk detector is used and there have been several such double-talk detectors proposed in the literature. A very simple such detector [5] that works when there is no near-end noise present computes the maximum cross-correlation between the input signal and the output signal and compares this to a threshold. Whenever this maximum cross-correlation exceeds a predefined threshold no double-talk is assumed to be present.

The room echo path

The impulse responses for the room echo paths are usually very long, and depending on the size of the room the reverberation time could be up to several hundred milliseconds. Of course, the reverberation time is in practice infinite but for practical purposes it can be considered to be finite. To model room impulse responses using FIR filters these filters need to be of lengths up to several thousand coefficients, depending on the sampling frequency.

Measures of echo cancellation performance

To measure the echo cancellation performance there are three measures that are usually used: the Echo Return Loss (ERL) [30], the Echo Return Loss Enhancement (ERLE) [5] and the misalignment [3]. Note that the signals and impulse responses used in these measures are explained in Figure 1.4.

The ERL is the ratio between the power of the returned echo and the power of the signal sent to the loudspeaker. It is measured in dB and is defined as

$$\mathrm{ERL} = -10\log_{10}\left(\frac{\sigma_y^2}{\sigma_x^2}\right) = -10\log_{10}\left(\frac{\mathrm{E}y^2(t)}{\mathrm{E}x^2(t)}\right) \tag{1.14}$$

where σ_y^2 and σ_x^2 are the powers of the echo signal and the input signal respectively.

The ERLE is a measure of the echo suppression achieved. It is defined by the ratio between the power of the residual echo and the power of the echo

$$\mathrm{ERLE} = -10\log_{10}\left(\frac{\sigma_e^2}{\sigma_y^2}\right) = -10\log_{10}\left(\frac{\mathrm{E}e^2(t)}{Ey^2(t)}\right) \tag{1.15}$$

where σ_e^2 is the power of the residual signal and σ_y^2 is the power of the microphone signal.

The misalignment is a measure of how well the estimated room impulse response models the "true" room impulse response. Note that the fact that the ERLE is low does not necessarily also mean that the misalignment is low. For a real application this measure is impossible to calculate, but usually, when testing AEC algorithms, estimated impulse responses are used for computing the echo data and thus we have access to the true impulse responses. These can then be used for calculating the misalignment which is defined as

$$\mathrm{M} = \frac{||\mathbf{h} - \tilde{\mathbf{h}}||^2}{||\tilde{\mathbf{h}}||^2}, \tag{1.16}$$

where $\tilde{\mathbf{h}}$ is the true room impulse response truncated to be of the same length as the estimated room impulse response $\mathbf{h}$.

1.1.4 Stereophonic acoustic echo cancellation

A variant of standard AEC that has become increasingly popular in recent years is Stereophonic Acoustic Echo Cancellation (SAEC) [2][5]. In teleconferencing systems it is desirable to have multiple loudspeakers and microphones to be able to transmit spatial sound information. This makes it possible to follow a panel of speakers and distinguish between different speakers by listening to the sound. Although the extension of the standard AEC algorithms to deal with this problem may seem to be relatively straightforward, the stereophonic setup introduces new problems that are extremely hard to solve.

The stereophonic acoustic echo canceller setup

A general stereophonic LEM system is depicted in Figure 1.5. As is seen it resembles the standard LEM system setup in Figure 1.1 to a large extent. What differs is that for SAEC there are four echo paths, two from each loudspeaker to each of the two microphones. Another difference is that since we have a stereophonic setup there are two input signals $x_1(t)$ and $x_2(t)$, and two output signals y_1 and y_2. Since the problem is symmetric for both microphones it is sufficient to deal with only one of them, the solution for the other microphone being similar.

The stereophonic LEM system is usually modeled as a MISO (Multiple Input Single Output) system

$$y_1(t) = h_{11} * x_1(t) + h_{12} * x_2(t) + n(t) \tag{1.17}$$

where $n(t)$ is noise, $*$ denotes convolution, and h_{11} and h_{12} are the room impulse responses to microphone M1 from speaker S1 and S2, respectively. This has to be done for both outputs so the computational complexity will be at least four times that for the standard case.

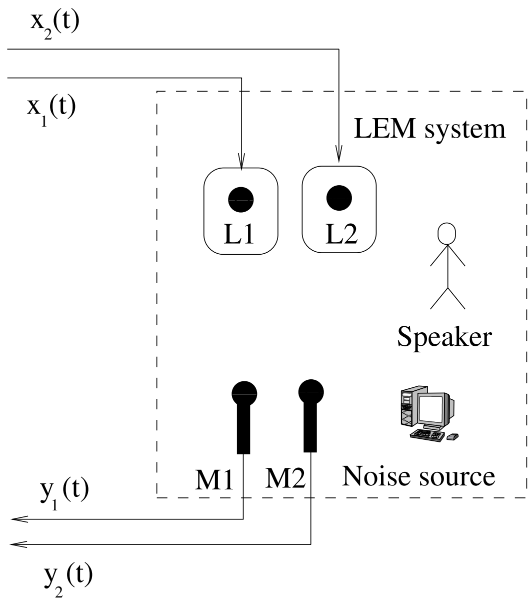

Figure 1.5: A stereophonic Loudspeaker-Enclosure-Microphone (LEM) system.

The nonuniqueness problem

The big problem with SAEC is that the estimation of h_{11} and h_{12} is highly ill-conditioned since $x_1(t)$ and $x_2(t)$ are highly correlated as they originate from the same source (the speaker in the far-end room), which is correlated in time since the source signal is speech. (This is especially true when the signal to noise ratio (SNR) is high since more independent, uncorrelated noise in $x_1(t)$ and $x_2(t)$ reduce the correlation.) The ill-conditioning will cause the problem of estimating $\mathbf{h}_{11}$ and $\mathbf{h}_{12}$ using standard techniques to have multiple solutions $\hat{\mathbf{h}}_{11}$ and $\hat{\mathbf{h}}_{12}$ and thus the estimates will generally not be equal to the true values of $\mathbf{h}_{11}$ and $\mathbf{h}_{12}$. Note that these estimates will anyway minimize the ERLE (the echo will be correctly estimated). The misalignment will, however, be high, which may not seem to be a problem since the echo is nevertheless cancelled. The problem is that the estimates of the room impulse responses are usually used only when they are not computed (that is, during double-talk) and therefore it is not sufficient that the impulse response estimates minimize the estimation error for the estimation of the echo signal only during the adaptation. It can be shown that the estimates of the impulse responses $\mathbf{h}_{11}$ and $\mathbf{h}_{12}$ in the near-end room are dependent on the impulse responses $\mathbf{g}_1$ and $\mathbf{g}_2$ between the speaker and the two microphones in the far-end room. This means that the algorithms will have to track changes in the echo paths of the far-end room as well as in those of the near-end room which is undesirable since the far-end impulse responses are highly time-varying (the speaker is moving or a new speaker starts talking). Acoustic echo cancellation relies on the assumption that the near-end room echo paths do not vary very much during double-talk. This cannot, however, be assumed for the echo paths of the far-end room. Therefore, it is important that the impulse response estimates of the near-end room echo paths equals the true impulse responses since in that case they are independent of the echo paths of the far-end room.

Approaches for solving the stereophonic echo cancellation problem

A large number of techniques have been proposed to solve the SAEC problem, none of which seems to be satisfactory. Many of the algorithms try to solve the SAEC problem by reducing the correlation of the input signals. This is feasible if the correlation can be reduced without adding too much audible distortion and without destroying the spatial audio information in the sound. One way to do this is to introduce a nonlinearity in one of the input signals [19]. Another is to regularize the correlation matrix (that is ill-conditioned) of the input signals by introducing leakage into the update of the coefficient vector [5].

A further more approach is to add independent random noise to the input channels [5]. A possible way to do this without disturbing the audible experience for the user would be to add noise that the human ear cannot hear, using psychoacoustics [5].

Another way to do the decorrelation has been to introduce complementary comb filters for the both channels [2], taking into account that, below 1 kHz, the frequencies are decisive for auditory localization [2]. Comb-filtering can be performed above this frequency and in the low frequency range a nonlinear transformation can be used to decorrelate the signals.

1.2 Sensor array signal processing

Sensor array signal processing deals with the problem of processing measurements from an array of sensors onto which signal waves are impinging, each of which are described by a finite number of parameters. This problem is of interest in many areas where wavefronts are involved. An example is in submarine localization equipment where several hydrophones are set up in a row and used for localizing possible submarines. Another is when electromagnetic signals from mobile phones are received using a collection of receiving antennas.

There are two basic kinds of problems to deal with. One is the detection problem that deals with finding the number of sources impinging on the array. The other is the estimation problem that deals with estimating the parameters for the different sources impinging on the array.

There have been a large amount of methods proposed for the problems of array signal processing but here we will only cover the basics of this field.

1.2.1 Sensor arrays

In Figure 1.6 a sensor array is depicted with a plane signal wave coming in from a direction (Direction Of Arrival (DOA)) θ (relative to the broadside of the array). In the figure the sensor array is arranged in a linear manner with equal distance between the sensors. This kind of array is called a Uniform Linear Array (ULA) and is the most common array variant. Another kind of array structure that is very common is the Non-Uniform Linear Array (NULA) where the sensors are arranged in a linear manner but not with equal spacing between the different sensors. There are also many other kinds of array configurations. Apart for the structure of the array other important properties of the array are the gain and phase responses of the array sensors.

Usually the array response is written in the form of the array steering vector (or the array response vector) $\mathbf{a}(\theta)$

$$\mathbf{a}(\theta) = \begin{bmatrix} H_1(\theta)e^{-j\mathbf{r}_1^T\mathbf{k}(\theta)} \\ H_2(\theta)e^{-j\mathbf{r}_2^T\mathbf{k}(\theta)} \\ \vdots \\ H_m(\theta)e^{-j\mathbf{r}_m^T\mathbf{k}(\theta)} \end{bmatrix} \tag{1.18}$$

where $\theta = [\psi\, \phi\, f]^T$ is the parameter vector containing the the frequency of the source and the DOAs (in two dimensions) of the source signal. $H_i(\theta)$ is

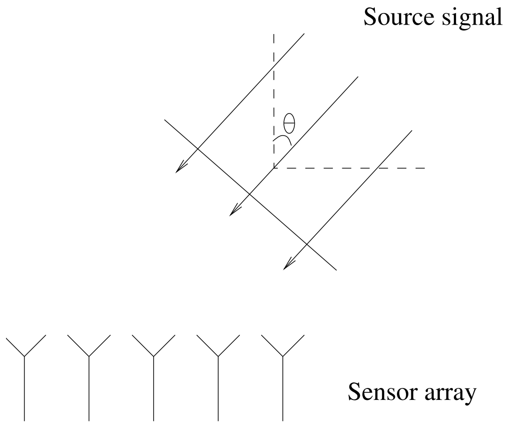

Figure 1.6: A 5 element sensor array with one signal impinging from direction θ.

the response of sensor i to a signal with parameter vector θ and the vector $\mathbf{r}_i = [r_x\, r_y\, r_z]$ describes the location of sensor i relative to a reference location. The wave-vector $\mathbf{k}(\theta)$ is defined as

$$\mathbf{k}(\theta) = \frac{2\pi f}{c}\left[-\cos(\psi)\cos(\phi) \quad -\sin(\psi)\cos(\phi) \quad -\sin(\phi)\right]^T, \tag{1.19}$$

where ψ, ϕ and f are defined in θ, and c is the propagation speed of the impinging wave. All possible array response vectors over the parameter range of interest form the so-called *array manifold*.

Note that in this model a crucial assumption is that the source is far-field which means that it is far enough away from the array for the wavefronts impinging on the array to be approximated as planar. Thereby the DOAs of a single source signal will be the same for all sensors in the array. If the sensor would be near-field this assumption would not be valid since the signal wavefronts impinging on the array would not be planar (but circular) and the DOAs for the signal wave at different sensors would not be the same.

1.2.2 Narrowband data model

The model for the array sensor outputs (for the narrowband case) is heavily dependent on the assumption that the signals dealt with are narrowband. Denote by $h_k(t)$ the impulse response of the kth sensor in the array, and assume that a real-valued continuous far-field signal $s(t)$ is impinging on the array and that τ_k is the delay of the signal at sensor k relative to some reference point. The output of the kth array sensor can then be written as

$$x_k(t) = h_k(t) * s(t - \tau_k) + n_k(t), \tag{1.20}$$

where $*$ denotes convolution and $n_k(t)$ is a noise term. The source signal $s(t)$ is usually written as

$$s(t) = \alpha(t)\cos(\omega t + \Phi(t)). \tag{1.21}$$

where ω is a center frequency. The assumption that the signal is narrowband means that the amplitude $\alpha(t)$ and the phase $\Phi(t)$ of the source signal varies slowly compared to the propagation time of the signal wave across the array. Then if τ is the propagation time for the source signal across the array we can use the approximation that $\alpha(t) \approx \alpha(t - \tau)$ and $\Phi(t) \approx \Phi(t - \tau)$ and we can write

$$\begin{aligned} s(t-\tau) &= \alpha(t-\tau)\cos(\omega(t-\tau) + \Phi(t-\tau)) \\ &\approx \alpha(t)\cos(\omega(t-\tau) + \Phi(t)). \end{aligned} \tag{1.22}$$

Hence we can write

$$\begin{aligned} x_k(t) &\approx h_k(t) * \alpha(t) cos(\omega(t-\tau_k) + \Phi(t)) + n_k(t) = \\ &= |H_k(i\omega)| a(t) cos(\omega(t-\tau_k) + \Phi(t) + arg(H_k(i\omega)) + n_k(t), \end{aligned} \tag{1.23}$$

where $H_k(i\omega)$ is the Fourier transform of $h_k(t)$. This means that the narrowband assumption allows us to to model the time delay of the signal as a simple phase shift of the carrier frequency.

Usually the signal $s(t)$ is written as a part of a complex signal $\bar{s}(t) = \alpha s(t) e^{i(\Phi(t)+\omega t)}$ denoted the analytic signal representation of $s(t)$ and we have that

$$s(t) = \frac{1}{2}(\bar{s}(t) + \bar{s}^H(t)). \tag{1.24}$$

Since the narrowband representation allows us to model a time delay as a phase shift we can write $\bar{s}(t-\tau_k) = e^{-iw\tau_k}\bar{s}(t)$ which means that

$$s(t-\tau_k) \approx \frac{1}{2}e^{-i\omega\tau_k}\bar{s}(t) + \frac{1}{2}e^{i\omega\tau_k}\bar{s}^H(t). \tag{1.25}$$

For high ω we have that $\bar{s}(t-\frac{\pi}{2\omega}) \approx \alpha(t)sin(\omega t + \Phi(t))$ and thereby we have that for large enough ω

$$s(t) \approx \bar{s}(t) + i\bar{s}(t - \frac{\pi}{2\omega}). \tag{1.26}$$

This construction of $\bar{s}(t)$ from discrete time measurements of $s(t)$ is called quadrature sampling. In the same way we can construct the complex valued signal $\bar{x}(t)$ from discrete measurements of $x(t)$ and in the same way we can also write the noise term $n(t)$ in a complex manner as $\bar{n}(t)$. Thus we can write

$$\bar{x}_k(t) = H_k(iw)e^{-iw\tau_k}\bar{s}(t) + \bar{n}_k(t), \tag{1.27}$$

which will be our basic model for the sensor array problem. From now on we will change notation, and by writing $s(t)$, $x(t)$ and $n(t)$ we will actually mean the complex signals in (1.27).

Using the model in (1.27) we arrive to the following parametrized data model for the sensor outputs

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ \vdots \\ x_m(t) \end{bmatrix} = \sum_{i=1}^{d} \mathbf{a}(\theta_i)s_i(t) + \mathbf{n}(t) = \mathbf{A}(\theta_{\mathbf{0}})\mathbf{s}(t) + \mathbf{n}(t), \tag{1.28}$$

where $\mathbf{x}(t)$ is a vector containing the sensor outputs, $\mathbf{a}(\theta_i)$ is the steering vector for source i, d is the number of sensors, $s_i(t)$ is the signal from source i, $\mathbf{A}(\theta_0)$ is a matrix whose column vectors are $\mathbf{a}(\theta_i)$, θ_0 is a vector containing all the different θ_i, $\mathbf{s}(t)$ is a vector containing all the source signals $s_i(t)$ and $\mathbf{n}(t)$ is a noise vector.

1.3 The sensor array problem

Observing the array output $x(t)$ at N time instants (the observations are called snapshots) the data matrix X_N for these snapshots can be formed as

$$X_N = [x(t_1) \ldots x(t_N)] = A(\theta_{\mathbf{0}})S_N + N_N, \tag{1.29}$$

where t_i denote sampling time i, and S_N and N_N are formed in the same way as X_N. The tasks are to estimate the number of signals d impinging on the array, to estimate the signal parameters θ_0 and to estimate the signals S_N using the array snapshot data matrix X_N and the model for the array response $A(\theta_0)$.

To do this we have to make some assumptions. One is that the array should be unambiguous, that is, that the array response should be linearly independent for any set of m signals with different parameter vectors. A necessary condition for this is that the model is identifiable which is fulfilled if the number m of signals impinging on the array is less than the number of sensors d (if some of the sources for the impinging signals are coherent the condition for unambiguity is different). Some usual assumptions about the noise process $n(t)$ is that it is to be zero-mean, stationary and white circularly complex Gaussian, which means that

$$E[n(t)n^H(s)] = \sigma^2 \delta_{t,s} I_m, \quad \text{and} \quad E[n(t)n^T(s)] = 0, \tag{1.30}$$

where E denotes the expectation value, $\delta_{t,s}$ the Kronecker delta and σ^2 the noise variance. The signals values $s(t)$ are often also assumed to be zero-mean, stationary and circularly complex Gaussian distributed random processes with second order moments

$$E[s(t)s^H(s)] = S\delta_{t,s}, \quad \text{and} \quad E[s(t)s^T(s)] = 0, \tag{1.31}$$

where S is the signal covariance matrix. Since $s(t)$ is usually assumed to be statistically independent of the noise $n(t)$ we can write the array output covariance matrix as

$$R_{xx} = E[x(t)x^H(t)] = A(\theta)SA^H(\theta) + \sigma^2 I_m. \tag{1.32}$$

The estimate of the covariance matrix R_{xx} is called the array sample covariance matrix and is formed as

$$\hat{R}_{xx} = \frac{1}{N}\sum_{t=1}^{N} x(t)x^H(t). \tag{1.33}$$

1.3.1 Different methods for array signal processing

A huge number of methods for performing array signal processing have been suggested over the years. Some of them will be summarized in this section.

The first class of methods is the spectral methods that use a beamformer $y(t) = w^H x(t)$ to form a beam in different directions and choose the DOA that maximizes the power of the beam output $y(t)$. In this beamformer the weight vector w can be constructed in different ways resulting in different methods, the most well-known being the conventional beamforming method [9] and Capon's method [6].

Then there are the parametric methods which consist of the Maximum Likelihood (ML) [17] methods for the case when the source signals are considered to be stochastic (Stochastic ML) [14] and the case when the source signals are considered to be deterministic (Deterministic ML) [4].

The subspace-based methods [39] are divided into two sub-classes. The first is the Signal Subspace Fitting (SSF) sub-class where the parameters for the array response vectors closest to the signal subspace (the subspace where the noise free signal $x(t) = A(\theta_{\mathbf{0}})s(t)$ is restricted to) is chosen. Depending on measures of "closeness" we get different methods. In the other sub-class, called Noise Subspace Fitting (NSF) the parameters corresponding to the array response vectors are chosen that are furthest from the noise subspace, which is orthogonal to the signal subspace. Many methods can be fitted into the subspace fitting framework, the most well-known being MUSIC [35]. Note that the spectral methods and the parametric methods can be fitted into this framework as well

Some methods are specialized for arrays with a certain structure, like Root-MUSIC [27] and IQML-MODE [36] that work on ULAs, and ESPRIT [29] that work when there are two separate, but otherwise identical arrays available.

1.3.2 Measures

An important measure for how well the methods for sensor array processing work is the covariance matrix of the estimation errors and there are several bounds on the elements of this that can be used to compare with to see how well the methods perform. Several lower bounds are available but the most commonly used is the Cramér-Rao lower Bound (CRB) [40]. One reason for this is that it is simple to use but also that it is often tight in the meaning that there exists an estimator that (asymptotically) achieves the CRB. If $\hat{\theta}$ is an unbiased estimate of the parameter vector $\theta_{\mathbf{0}}$ ($E(\hat{\theta}) = \theta_0$) based on the observations X_N the CRB on the covariance of the estimation error is given by

$$E[(\hat{\theta}-\theta_0)(\hat{\theta}-\theta_0)^T] \geq \left[-E\left\{\frac{\partial^2 \log(X_N|\theta)}{\partial\theta\partial\theta^T}\right\}\right]^{-1}. \tag{1.34}$$

1.3.3 The detection problem

There are several different techniques for detecting the number of source signals impinging on the array. Some methods have been proposed that work by testing the multiplicity of the smallest eigenvalue of $\hat{R}_{xx}$ [31]. Since the lowest eigenvalue corresponds to the noise the number of signals present can be calculated from the eigenvalues. These methods, however, only work for detecting the number of non-coherent source signals. Other approaches that work for arbitrary emitter signals are the Generalized Likelihood Ratio Test (GLRT) [33] and the Minimum Description length (MD) principle [28].

1.3.4 Near-field

In the sensor array model above the sources have all been assumed to be far-field which means that the waves coming from the sources can be approximated as planar over the array. In this way the delay of the signals at the different sensors does not depend on the distance to the source (and neither does the DOA of the source signal). If the source is in the near-field of the array this is, however, no longer a valid assumption and the signal waves reaching the array are circular and the delays at the different sensors are dependent on the distance as well as the direction to the source. To cope with this the model for the array response vectors has to be modified and the distance to the source has to be introduced as a parameter in θ.

1.3.5 Wideband

If the source signals are wideband the narrowband approximation is no longer valid and the narrowband techniques cannot be applied directly. There is always the option to filter the data using passband filters with narrow passbands and apply the techniques discussed above on the filtered data, but a more unified solution is generally desired. One method dealing with this is presented in [10].

1.3.6 Beamforming

Another application for sensor arrays is beamforming [24][23], in which the sensor array is used to focus the array in a certain direction (usually towards a source) in order to enhance the SNR of the signal in that direction. This enhancement is accomplished by:

- Reducing the sensor measurement noise since several independent sensors are used for estimating the signal.
- Reducing the interference from other sources (which could be noise sources) by suppressing signals coming from directions other than the look direction.

A typical beamformer is depicted in Figure 1.7 where θ is the DOA of an

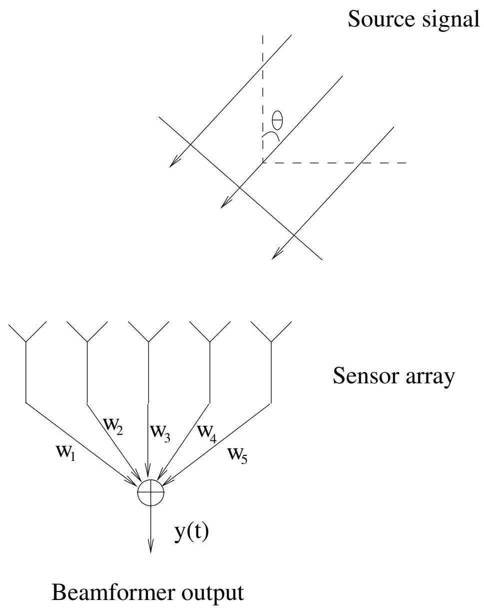

Figure 1.7: A beamformer using a 5 element sensor array with one signal impinging from direction θ.

impinging source signal and w_i are the beamformer weights for the array sensor outputs. $y(t)$ is the beamformer output.

There are two basic types of beamformers. The first one is data-independent beamformers that have a fixed beampattern (fixed beamformer weights) optimized in a certain look direction according to different criteria, independent of the assembled array data. A drawback with these is that beampattern cannot be optimized to maximally suppress interfering sources impinging on the array. The data-independent beamformers are, however, very computationally efficient since no optimization has to be done during processing to calculate the optimal beamformer based on the collected array output data. These kinds of beamformers are therefore often found in applications where the processing power available is extremely limited. A typical example is hearing aids.

The other type of beamformer is the data-dependent beamformer that optimizes the beampattern (the beamformer weights) according to different criteria using the collected array output data. This enables the data-dependent beamformer to concentrate the degrees of freedom in the optimization to maximally suppress interfering source signals impinging on the array and the beamformer can apply the suppressing power where it is most needed. This is the kind of beamformers that are most commonly used.

As in the estimation problem for array signal processing special approaches have to be used for beamforming when the source signal cannot be considered to be narrowband [22], and when the source cannot be considered to be in the far-field of the array [21].

1.4 Contributions

In this section the papers I have written up until now are listed. A brief description of each of them is given.

1. "On high-resolution frequency analysis with a small data record", Per Åhgren and Petre Stoica. Published in *Electronics Letters* , Vol. 36, No. 20, Sept. 2000. Not included in the thesis. The paper applies the nonlinear LS algorithm to the problem of frequency analysis using a small data record.

2. "Exact initialization of the recursive least-squares algorithm", Petre Stoica and Per Åhgren. Accepted for publication in *Journal of Adaptive Control and Signal Processing*, 2001. This is included in the thesis as chapter 4. The paper presents a version of RLS with exact initialization

in the sense that it yields exactly the same estimates as LS. This is achieved without an increase in computational complexity.

3. "Echo Cancellation using the Conjugate Gradient Algorithm", Per Åhgren and Erik G. Larsson. To be submitted to *Journal of Adaptive Control and Signal Processing*, 2001. This is included in the thesis as chapter 2. The paper presents and applies a new method for filter adaptation to the problem of acoustic echo cancellation in both stereo and mono.

4. "Echo Cancellation using the Conjugate Gradient Algorithm", Per Åhgren and Erik G. Larsson. Presented at the *International Workshop of Acoustic Echo and Noise Cancellation* in Darmstadt, Germany, September 10-13, 2001. Not included in the thesis. The paper is a conference version of the previous paper.

5. "Array Processing for Signals with Non-Zero Means in Colored Noise Fields", Petre Stoica, Monika Agrawal and Per Åhgren. Submitted to *IEEE Transactions on Signal Processing*, 2001. This is included in the thesis as chapter 7. The paper considers the array signal processing problem for signals with non-zero means in colored noise. A ML method is derived for both the estimation and the detection problems and a statistical analysis is provided for the ML method for the signal parameter estimation. Furthermore an expression is derived for the CRB matrix associated with the problem.

6. "Common Factor Estimation and Two Applications in Signal Processing", Monika Agrawal, Petre Stoica and Per Åhgren. Submitted to *Signal Processing*, 2001. This is included in the thesis as chapter 5. The paper solves the problem of extracting common factors from a set of polynomials. Two problems of array processing and system identification are posed as common factor estimation problems and solved using the proposed method.

7. "Data-dependent broad-band beamforming with many constraints using few sensors", Per Åhgren and Monika Agrawal. To be submitted, 2001. This is included in the thesis as chapter 6. The paper considers the problem of data-dependent beamforming when there are few sensors in the array. A method is proposed where the beamformer

constraints are fulfilled approximately instead of exactly. Hence, additional degrees of freedom become available for the beamformer to suppress noise sources.

8. "On the hierarchical RLS algorithm", Petre Stoica, Monika Agrawal and Per Åhgren, Submitted to *IEEE Communications Letters*, 2001. This is included in the thesis as chapter 3. The paper provides an analysis of the hierarchical RLS algorithm and sheds more light on the results presented in previous papers for this algorithm.

9. "Estimation of Nominal Directions of Arrival and Angular Spreads of Distributed Sources", Monika Agrawal, Petre Stoica, Olivier Besson, Per Åhgren. Submitted to *Signal Processing*, 2001. Not included in the thesis. In the paper the array signal processing problems of DOA and angular spread estimation for distributed sources are considered. A computational efficient method is proposed for the multiple source case for ULA configurations. In the numerical examples it is shown to achieve the CRB.

10. "Optimum beamforming for a near-field source", Monika Agrawal, Per Åhgren and Richard Abrahamsson. Under processing, 2001. Not included in the thesis. The paper deals with the beamforming problem for the case when the desired source is nearfield, but the interfering sources are far-field. The results in the paper makes it possible to obtain new near-field beamformers by generalizing existing far-field beamformers.

1.5 Future research

There are several possibilities for future research that can be attempted in the framework of this thesis. Some of these are:

1. For high quality audio applications a solution to the illconditioning problem of stereophonic acoustic echo cancellation has to be obtained. Current methods rely on decorrelation of the input signals which will not be feasible for high quality audio applications since the decorrelation reduces the quality of the audio signals.

2. For beamforming in teleconferencing the introduction of available knowledge about the teleconferencing setup into the beamformer optimization problem can be studied. By allowing available knowledge

about the teleconferencing setups, such as common speaker placement and common locations of the interferers, to be used in the optimization of the array algorithms for beamforming it should be possible to get new, better performing beamformers.

3. Faster and computationally less demanding algorithms are desired for the problem of acoustic echo cancellation, especially in the stereophonic case.

4. The echo path imbalance problem in stereophonic acoustic echo cancellation is a very interesting problem. If directional microphones are used in the stereophonic setup the two echo paths will be very different in magnitude and thereby the one with the lesser magnitude will be very sensitive to errors. This imbalance problem has to be dealt with.

1.6 Abbreviations and acronyms

AEC	Acoustic Echo Cancellation
AP	Affine Projection
CRB	Cramér-Rao lower Bound
DOA	Direction Of Arrival
ERL	Echo Return Loss
ERLE	Echo Return Loss Enhancement
FAP	Fast Affine Projection
FIR	Finite Impulse Response
FTF	Fast Transversal Filter
GLRT	Generalized Likelihood Ratio Test
IIR	Infinite Impulse Response
ITU	International Telecommunication Union
LEM	Loudspeaker-Enclosure-Microphone
LMS	Least Mean Squares
LS	Least Squares
MD	Minimum Description length
MISO	Multiple Input Single Output
ML	Maximum Likelihood
NLMS	Normalized Least Mean Squares
NLP	Non-Linear Processor
NSF	Noise Subspace Fitting
NULA	Non-Uniform Linear Array
RLS	Recursive Least Squares
SAEC	Stereophonic Acoustic Echo Cancellation
SAP	Sensor Array Processing
SNR	Signal to Noise Ratio
SSF	Signal Subspace Fitting
ULA	Uniform Linear Array

Bibliography

[1] Allen JB, Berkley DA. Image method for efficiently simulating small-room acoustics. *J. Acoust. Soc, Am.*, 65:943–950, 1979.

[2] Benesty J, Morgan DR, Hall JL, Sondhi MM. Stereophonic acoustic echo cancellation using nonlinear transformations and comb filtering. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing 1998*, volume 6, pages 3673–3676, 1998.

[3] Benesty J, Morgan DR, Sondhi MM. A hybrid mono/stereo acoustic echo canceler. *IEEE Transactions on Speech and Audio Processing*, 6(5):468–475, Sept 1998.

[4] Böhme JF. Estimation of Source Parameters by Maximum Likelihood and Nonlinear Regression. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing 1984*, pages 7.3.1–7.3.4, 1984.

[5] Breining C, Dreiseitel P, Hänsler E, Mader A, Nitsch B, Puder H, Schertler T, Schmidt G, Tilp J. Acoustic Echo Control - An Application of Very-High-Order Adaptive Filters. *IEEE Signal Processing Magazine*, 16(4):42–69, July 1999.

[6] Capon J. High Resolution Frequency Wave Number Spectrum Analysis. *Proceedings of IEEE*, 57:1408–1418, 1969.

[7] Chen WY, Dixon JL, Waring DL. High Bit Rate Digital Subscriber Line Echo Cancellation. *IEEE Journal on Selected Areas in Communications*, 4(6):848–860, Aug 1991.

[8] Dreiseitel P, Hänsler E, Puder H. Acostic echo and noise control - a long lasting challenge. In *Proc. EUSIPCO-98, 9th European Signal Processing Conference*, Sept 1998.

[9] Dudgeon DE. Fundamentals of digital array processing. *Proceedings of the IEEE*, 65:898–904, June 1977.

[10] Frikel M, Boutennane S. Statistical analysis to wideband array processing. In *Proceedings of IEEE TENCON '96 Digital Signal Processing Applications*, volume 1, pages 435–439, 1996.

[11] Glentis GO, Berberidis K, Theodoridis S. Efficient Least Squares Adaptive Algorithms For FIR Transversal Filtering. *IEEE Signal Processing Magazine*, 16(4):13–41, July 1999.

[12] Hänsler E. The hands-free problem - An annotated bibliography. *Signal Processing*, 3(27):259–271, 1992.

[13] Haykin S. *Adaptive Filter Theory.* Prentice-Hall, Upper Saddle River, New Jersey, 3rd edition, 1996.

[14] Jaffer AG. Maximum Likelihood Direction Finding of Stochastoc Sources: A separable Solution. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing 1988*, volume 5, pages 2893–2896, Apr 1988.

[15] Liavas AP, Regalia PA. Acoustic echo cancellation: do IIR models offer better modeling capabilities than their FIR counterparts. *IEEE Transactions on Signal Processing*, 46(9):2499–2504, Sept 1998.

[16] Ljung L, Söderström T. *Theory and Practice of Recursive Identification.* MIT Press, 1983.

[17] Mardia KV, Kent JT, Bibby JM. *Multivariate Analysis.* Academic Press, London, 1979.

[18] Mboup M, Bonnet M. On the adequateness of IIR adaptive filtering for acoustic echo cancellation. In *Proc. EUSIPCO-92, 6th European Signal Processing Conference*, volume 1, pages 111–114, Sept 1992.

[19] Morgan DR, Hall JL, Benesty J. Investigation of several types of nonlinearities for use in stereo acoustic cancellation. *IEEE Transactions on Speech and Audio Processing*, 9(6):686–696, Sept 2001.

[20] Ng MK, Plemmons RJ. Fast Recursive Least Squares Adaptive Filtering by Fast Fourier Transform-based Conjugate Gradient Iterations. *SIAM Journal of Scientific Computing*, 17(4):920–941, July 1996.

[21] Nordebo S, Claesson I, Nordholm S. A minimax near field design of broadband beamformers. In *Antennas and Propagation Society International Symposium, 1994. AP-S. Digest , 1994*, volume 1, pages 120–123, 1993.

[22] Nordebo S, Claesson I, Nordholm S. Broadband Adaptive Beamforming: A design Using 2-D Spatial Filters. In *Antennas and Propagation Society International Symposium, 1993. AP-S. Digest , 1993*, volume 2, pages 702–705, 1993.

[23] Nordebo S, Claesson I, Nordholm S. Adaptive beamforming: Spatial filter designed blocking matrix. *IEEE Journal of Oceanic Engineering*, 19(4):583–590, Oct 1994.

[24] Nordholm SE, Rehbock V, Tee KL, Nordebo S. Adaptive beamforming: Spatial filter designed blocking matrix. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 45(1):141–143, Jan 1998.

[25] Ozeki K, Umeda T. An Adaptive Filtering Algorithm Using an Orthogonal Projection to an Affine Subspace and Its Properties. *Electronics and Communications in Japan*, 67-A(5):19–27, 1984.

[26] Proakis J, Manolakis D. *Digital Signal Processing.* Prentice-Hall International Editions, 1996.

[27] Rao BD, Hari KVS. Performance analysis of Root-MUSIC. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(12):1939–1949, Dec 1989.

[28] Rissanen J. Modeling by Shortest Data Description. *Automatica*, 14:465–471, 1978.

[29] Roy R, Paulraj A, Kailath T. ESPRIT - A Subspace Rotational Approach to Estimation of Parameters of Cisoids in Noise. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 34(4):1340–1342, Oct 1986.

[30] Sankaran SG, Beex AA. Acoustic echo and noise canceler improvements for hands-free telephones. In *Proceedings of Southeastcon '97. Engineering new New Century. IEEE.*, pages 148–150, Apr 1997.

[31] Schreiber R. Implementation of Adaptive Array Algorithms. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 34:1038–1045, Oct 1986.

[32] Schütze H, Ren Z. Numerical characteristics of fast recursive least squares transversal adaptation algorithms - A comparative study. *Signal Processing*, 23(3):317–322, June 1992.

[33] Silvey SD. *Statistical Inference.* Penguin Books, London, 1970.

[34] Stoica P, Moses R. *Introduction to Spectral Analysis.* Prentice-Hall, Upper Saddle River, New Jersey, 1997.

[35] Stoica P, Nehorai A. MUSIC, Maximum Likelihood and Cramér-Rao Bound. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37:720–741, May 1989.

[36] Stoica P, Sharman KC. Maximum likelihood methods for direction-of-arrival estimation. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38:1132–1143, July 1990.

[37] Tahernezhadi M, Sosale D. Performance evaluation of a DSP-based teleconferencing system. *IEEE Transactions on Consumer Electronics*, 40(4):957–962, Nov 1994.

[38] The new Bell telephone. *Scientific American*, 37:1, 1877.

[39] Viberg M, Ottersten B. Sensor Array Processing Based on Subspace Fitting. *IEEE Transactions on Signal Processing*, 39(5):1110–1121, May 1991.

[40] Weiss AJ, Weinstein E. Lower Bounds in Parameter Estimation - Summary of Results. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing 1986*, pages 569–572, 1986.

Chapter 2

Echo Cancellation using the Conjugate Gradient Algorithm

Echo Cancellation using the Conjugate Gradient Algorithm

Abstract

In this paper an approximation to the sliding window Recursive Least Squares (RLS) algorithm with filter estimate updates using the conjugate gradient algorithm is applied to the Acoustic Echo Cancellation (AEC) problem for both the mono and stereo cases. The method is shown to perform much better than the Normalized Least Mean Squares (NLMS) algorithm which is one of the standard algorithms used for AEC today. While it performs somewhat worse than RLS which is the optimal (unbiased) estimator in a least squares sense, it is shown to be computationally much less demanding. In contrast to RLS we believe that a real-time implementation of our algorithm for AEC can be feasible and hence the method presented here should be one of the best algorithms to consider for the AEC problem.

KEY WORDS: Acoustic Echo cancellation; Low complexity RLS approximation; Conjugate gradient algorithm iterations; Stereo and Mono

2.1 Introduction and problem formulation

In almost all communication paths echoes are present. Depending on the delay of the echo paths these may be noticeable or not to the user. If the delay exceeds a few tenths of milliseconds the echoes will be noticeable and can be quite annoying. In hands-free telephony [8] the echo delay is often that long due to the delay caused by the telephone network and due to the hands-free communication setup. A simple solution to remove the echoes in hands-free communication would be to use *half-duplex* communication, that is, allow communication only in one direction at one time. This works fine but does not allow for double-talk [13], which occurs when communication in both directions occur simultaneously (i.e. when both speakers are talking at the same time). To allow double-talk without having echoes disturbing the communication, acoustic echo cancellation (AEC) is needed.

The acoustic echoes arise from the acoustic coupling between the loudspeaker(s) and the microphone(s) in the hands-free setup. There are in principle two different kinds of setups: the monophonic setup and the stereophonic setup [3], of which the former is a special case of the latter. The stereophonic setup is illustrated in Figure 2.1. The speech from the far end is picked up by

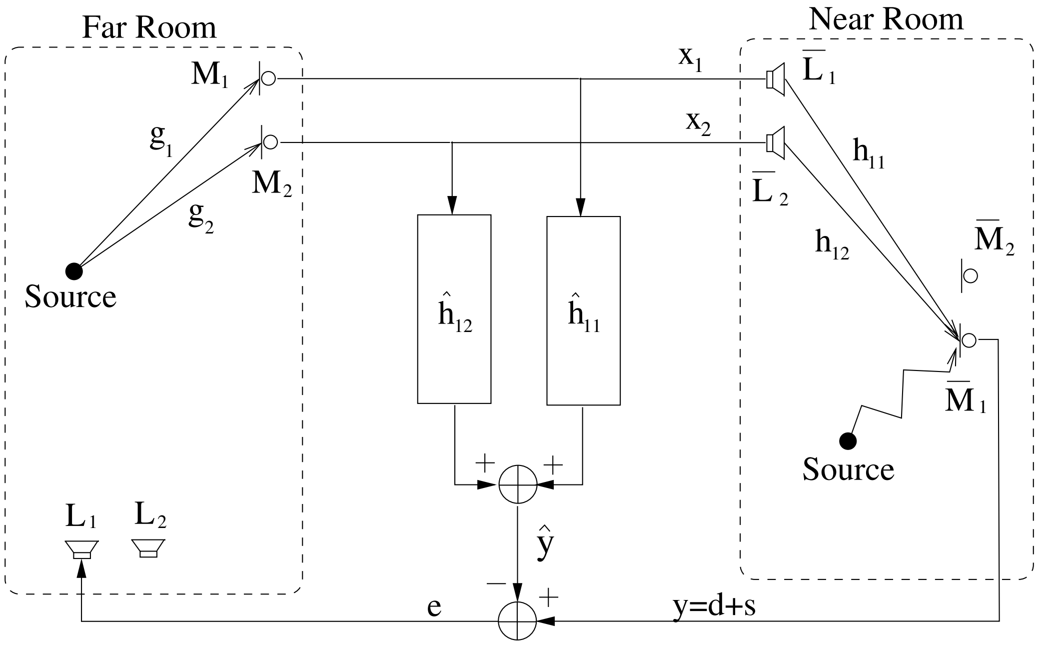

Figure 2.1: Stereophonic acoustic echo cancellation.

the microphones M_1 and M_2, and is sent to the loudspeakers L_1 and L_2 as the signals $x_1(t)$ and $x_2(t)$. It then propagates through the near-end room and is picked up by the microphones $\bar{M}_1$ and $\bar{M}_2$ as the undesired echo signals that should not be propagated back to the far-end room. Note that the problem is symmetric in the sense that the process of removing the echo from one microphone signal is equivalent to removing the echo from the other, and thus only the microphone $\bar{M}_1$ will be treated. Furthermore there may be near-end speech (in the near-end room) which is also picked up by the microphone $\bar{M}_1$ as the desired signal $s(t)$ that should be transmitted to the loudspeaker L_1. The resulting microphone signal picked up by $\bar{M}_1$ is thus $y(t) = d(t) + s(t)$, where $d(t)$ is the undesired echo signal. As long as $s(t) \equiv 0$, the undesired signal $d(t)$ can simply be removed by setting $y(t) \equiv 0$, and when there is no sound from the loudspeakers $\bar{L}_1$ and $\bar{L}_2$, $x_1(t) \equiv x_2(t) \equiv 0$ and there is no echo. The problem arises, however, when both $s(t) \neq 0$ and $d(t) \neq 0$ since we then have *full-duplex communication* (double-talk). Then the echo signal $d(t)$ has to be removed by stereophonic acoustic echo cancellation (SAEC), using two estimated filters (usually FIR filters) that approximately model the echo paths from the loudspeakers to the microphone in the near end room (the room impulse responses $\mathbf{h}_{11}(t)$ and $\mathbf{h}_{12}(t)$). The filters are used to estimate the undesired echo signal $d(t)$ which is subtracted from the microphone signal $y(t)$, thereby removing the echo signal. Usually, the filters can only be updated when $x_1(t) \neq 0$, $x_2(t) \neq 0$ and $s(t) \equiv 0$ (echo is present but no double-talk).

The monophonic setup is similar to the stereophonic setup except for that only one loudspeaker and one microphone in each room are used and only one filter is required to model the echo path. The echo is then removed by monophonic acoustic echo cancellation (MAEC).

The estimation algorithm for the filter modeling the room impulse responses has to fulfill the following important requirements:

1. The room impulse responses have proven to be very long in general, and for a standard office room with a hands-free setup working with a sampling frequency of 16 kHz they can have a length of several thousand filter taps (in the time domain) [3]. Thus it is very important that the algorithm is not too computationally complex and that it does not require the storage of too large amounts of data, since it must be feasible to implement on a standard Digital Signal Processor (DSP).

2. Since the room impulse responses can change very rapidly, for instance when a person moves around in the near-end room, it is very important for the algorithm to adapt rapidly to changes. Note, however, that the adaptation must not be too fast. Since adaptation is usually only allowed when there is no double-talk, a double-talk detector must be used to detect when to turn on and off the adaptation. As most double-talk detectors do not detect double-talk instantaneously, the algorithm will adapt during double-talk for a short time period which in turn will cause the filter estimates to deteriorate. If the adaptation is slow this will not be a problem but if it is very fast the quality of the filter estimates could decline significantly before the adaptation is turned off.

3. Since the estimates of the room impulse responses are usually used only when they are not computed (that is, during double-talk) it is not sufficient that the impulse response estimates minimize the estimation error

for the estimation of the echo signal only during the adaptation. The important thing is to minimize the estimation error when the impulse response estimates are used and since most algorithms do not allow filter estimate updates during double-talk (when AEC is needed), it is also important to minimize the misalignment [2], which is the difference between the true room impulse response and the estimated room impulse response. Usually this is not a problem in the monophonic case. In the stereophonic case, though, it is a difficult problem since the signals $x_1(t)$ and $x_2(t)$ are highly correlated and thus the problem of estimating the filters becomes extremely ill-conditioned (and in general almost does not have a unique solution). In fact it can be shown that all (nonunique) solutions to the stereophonic problem (except for the solution equal to the true room impulse responses) depend on the echo paths of the far-end room as well as those of the near-end room [5]. Thus if nothing is done to reduce the correlation between $x_1(t)$ and $x_2(t)$ (and thereby make the problem less ill-conditioned), the algorithm used to cancel the echo has to respond to all the changes in the echo paths of the far-end room and if, for instance, one speaker stops talking in the far-end room and another speaker starts talking at another location in the far-end room, the algorithm has to reconverge to new (nonunique) impulse response estimates. Since there is no algorithm that can respond to such changes sufficiently fast this is a major problem in SAEC.

A vast number of methods have been tested for the MAEC/SAEC problem [4] of which probably the most commonly used method, due to its low computational complexity, is the normalized least mean squares (NLMS) [9] algorithm. It adapts, however, very slowly and what would be desired is an algorithm that adapts faster such as the recursive least squares (RLS) algorithm [10]. The RLS[1] algorithm is, however, too computationally complex and requires too much memory to be implemented in a real-time application on a standard DSP. Instead algorithms such as the fast recursive least squares (FRLS) [9] and the affine projection (AP) algorithm [12] have been proposed. The FRLS algorithms suffer, however, from problems with numerical instability and it has been shown [11] that all known FRLS algorithms are unstable when using a forgetting factor less than one (which is required

[1]Unless otherwise stated, we mean by RLS the (exact) recursive solution to the least squares problem [14].

to diminish the effects of old data to allow rapid adaptation to changes). The AP algorithm is inbetween the RLS and NLMS algorithms both regarding complexity and speed of adaptation and can trade increased adaptation speed against increased computational complexity.

A common approach to reduce the ill-conditioning in SAEC is to introduce a nonlinearity in the loudspeaker signals, thereby effectively decorrelating these and providing a unique solution for the SAEC problem. Several different choices of nonlinearities to use have been proposed and the choice of nonlinearity that has perhaps yielded the greatest response is the half-wave rectifier that was successfully applied to the SAEC problem in [2].

In [11] a low complexity approximation of RLS with sliding data window was proposed which we will call Conjugate-Gradient RLS (CG-RLS). CG-RLS was shown to have a very fast speed of adaptation while requiring a relatively small amount of memory and being computationally much less demanding than the RLS algorithm. The goal of this paper is to apply the technique in [11] to the AEC problem. We also discuss some practical issues such as speed of adaptation versus the number of iterations, window- and filter lengths, and computational complexity.

2.2 Conjugate-Gradient RLS

The sliding window RLS algorithm tries to estimate a FIR filter $\mathbf{h}(t)$ that minimizes the sum of squared errors

$$V(t) = \sum_{j=t-M+1}^{t} |e(j)|^2 \triangleq \sum_{j=t-M+1}^{t} |d(j) - \mathbf{h}^*(t)\mathbf{x}(t)|^2 \triangleq \|\mathbf{d}(t) - \mathbf{X}(t)\mathbf{h}(t)\|^2, \tag{2.1}$$

where $\mathbf{h}(t)$ is the filter to estimate, $\mathbf{x}(t) \triangleq [x(t)\ x(t-1)\ \ldots\ x(t-n+1)]^T$ is the input data vector, $d(j)$ is the known desired filter output, $\mathbf{d}(t) \triangleq [d(t-M+1)\ d(t-M+2)\ \ldots\ d(t)]^T$, n is the length of the filter $\mathbf{h}(t)$ and M is the length of the sliding data window. The Mxn Toeplitz matrix $\mathbf{X}(t)$

is defined as

$$\mathbf{X}(t) \triangleq \begin{bmatrix} \overline{x(t-M+1)} & \dots & \dots & \overline{x(t-M-n+2)} \\ \vdots & \ddots & & \vdots \\ \overline{x(t-n+1)} & & \ddots & \vdots \\ \vdots & \ddots & & \overline{x(t-M+1)} \\ \vdots & & \ddots & \vdots \\ \overline{x(t)} & \dots & \dots & \overline{x(t-n+1)} \end{bmatrix}, \tag{2.2}$$

where $\bar{(\cdot)}$ denotes the complex conjugate. As is well-known, the filter $\mathbf{h}(t)$ minimizing (2.1) is given by the solution to the normal equations

$$\mathbf{X}^*(t)\mathbf{X}(t)\mathbf{h}(t) = \mathbf{X}^*(t)\mathbf{d}(t), \tag{2.3}$$

which can be computed in a number of different ways. The perhaps most common approach is to form a recursion of $(\mathbf{X}^*(t)\mathbf{X}(t))^{-1}$ and $\mathbf{X}^*(t)\mathbf{d}(t)$ in time, and solve for $\mathbf{h}(t)$ by $\mathbf{h}(t) = (\mathbf{X}^*(t)\mathbf{X}(t))^{-1}\mathbf{X}^*(t)\mathbf{d}(t)$ as is done by RLS [10], [11]. The approach taken in [11] is to solve (2.3) by using the conjugate gradient (CG) algorithm [7] which can be efficiently implemented for the least squares problem at hand using the Fast Fourier Transform (FFT) and its inverse (IFFT). This is an algorithm that iteratively solves (2.3) exactly from any initial estimate in at most as many steps as the number of parameters to estimate, i.e., n. If the condition number of $\mathbf{X}^*(t)\mathbf{X}(t)$ is, low, fewer steps are required. Also, if the initial estimate is good then CG will converge much faster to the solution. The method proposed in [11] is to solve (2.3) using CG, at time t using the solution obtained at time $t-1$ as an initial estimate. In that way only one or a few iterations per time step are required to ensure good adaptation properties of the algorithm.

The CG algorithm is presented in [7] but for completeness we present it here as well using the terminology used for our problem. Denoting $\mathbf{R}(t) \triangleq \mathbf{X}^*(t)\mathbf{X}(t)$ and $\mathbf{p}(t) \triangleq \mathbf{X}^*(t)\mathbf{d}(t)$ we describe the CG algorithm for solving (2.3) in Table 1. Note that in the CG algorithm in Table 1 the initial filter estimate $\mathbf{h}(t)$ is set as the filter estimate of the previous time step as was done in [11]. Also note that the most computational demanding part in the algorithm is computing $\mathbf{R}(t)\mathbf{q}$, which for the CG algorithm to be computationally attractive must be done using as small a number of operations as possible. As we will see this matrix-vector product can be performed very efficiently

$k = 0$
$\mathbf{r} = \mathbf{p}(t) - \mathbf{R}(t)\mathbf{h}(t-1)$
$\rho_0 = \|r\|_2^2$
While $(\sqrt{\rho_k} > \epsilon\|\mathbf{p}(t)\|_2$ and $k < k_{MAX})$
  $k = k + 1$
  If $k = 1$
    $\mathbf{q} = \mathbf{r}$
  else
    $\beta_k = \rho_{k-1}/\rho_{k-2}$
    $\mathbf{q} = \mathbf{r} + \beta_k \mathbf{q}$
  $\mathbf{w} = \mathbf{R}(t)\mathbf{q}$
  $\alpha_k = \rho_{k-1}/\mathbf{q}^T\mathbf{w}$
  $\mathbf{h}_k(t) = \mathbf{h}_{k-1}(t) + \alpha_k \mathbf{q}$
  $\mathbf{r} = \mathbf{r} - \alpha_k \mathbf{w}$
  $\rho_k = \|\mathbf{r}\|_2^2$
$\mathbf{h}(t) = \mathbf{h}_k(t)$

Table 2.1: The conjugate gradient algorithm.

using the FFT/IFFT. First note that $\mathbf{R}(t) = \mathbf{X}^*(t)\mathbf{X}(t)$ and that $\mathbf{X}(t)$ (as well as $\mathbf{X}^*(t)$) is a Toeplitz matrix. It is well known that any Toeplitz matrix $\mathbf{X}(t)$ can be extended to a circulant matrix $\mathbf{C} = \begin{bmatrix} \mathbf{X}(t) & \mathbf{E} \\ \mathbf{G} & \mathbf{H} \end{bmatrix}$, where $\mathbf{E}$, $\mathbf{G}$ and $\mathbf{H}$ are suitably chosen matrices. Since circulant matrices can be decomposed as $\mathbf{C} = \mathbf{F}^*\mathbf{D}\mathbf{F}$ where $\mathbf{F}$ is the Fourier (or Discrete Fourier Transform) matrix and $\mathbf{D}$ is a diagonal matrix whose elements are the FFT of the first column of $\mathbf{C}$ (note that the dimension of the square matrix $\mathbf{C}$ can easily be adjusted to be a power of two so that the FFT can be applied). Thus the multiplication $\mathbf{X}(t)\mathbf{q} \triangleq \mathbf{a}$ can be computed as $\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} = \mathbf{C}\begin{bmatrix} \mathbf{q} \\ \mathbf{0} \end{bmatrix} = \mathbf{F}^*\mathbf{D}\mathbf{F}\begin{bmatrix} \mathbf{q} \\ \mathbf{0} \end{bmatrix}$, where the value of $\mathbf{b}$ is unimportant. Since multiplying a Fourier matrix with a vector is an IFFT operation (or a FFT operation for the inverse of the Fourier matrix) the multiplication $\mathbf{X}(t)\mathbf{q}$ can be performed using the FFT/IFFT. The multiplication $\mathbf{X}^*(t)(\mathbf{X}(t)\mathbf{q}) \triangleq \mathbf{R}(t)\mathbf{q}$ can be computed in a similar manner using the FFT/IFFT.

We remark that the CG algorithm is often used with a *preconditioner*, leading to the so-called preconditioned CG (PCG) algorithm. This amounts to solve the transformed system $\widetilde{\mathbf{X}^*(t)\mathbf{X}(t)}\widetilde{\mathbf{h}(t)} = \widetilde{\mathbf{X}^*(t)\mathbf{d}(t)}$, where $\widetilde{\mathbf{X}^*(t)\mathbf{X}(t)} \triangleq \mathbf{P}^{-1}(t)\mathbf{X}^*(t)\mathbf{X}(t)\mathbf{P}^{-1}(t)$, $\widetilde{\mathbf{X}^*(t)\mathbf{d}(t)} \triangleq \mathbf{P}^{-1}(t)\mathbf{X}^*(t)\mathbf{d}(t)$, $\widetilde{\mathbf{h}(t)} \triangleq \mathbf{P}(t)\mathbf{h}(t)$ and the preconditioner $\mathbf{P}(t)$ is chosen such that $\widetilde{\mathbf{X}^*(t)\mathbf{X}(t)}$ has a low condition number. Since it appears for our problem that the CG algorithm works well also without preconditioning, and since preconditioning adds considerably to the computational complexity we do not consider it in this work.

2.3 CG-RLS applied to MAEC and SAEC

2.3.1 Issues regarding implementation and computational complexity

As mentioned in Section 1 the allowed computational complexity and storage requirements are limited in an implementation of an AEC/SAEC algorithm. For CG-RLS the storage requirements are of the order $\mathcal{O}(n+M)$ since only the first column and first row of the matrix $\mathbf{X}(t)$ are used for the fast multiplication $\mathbf{X}^*(t)\mathbf{X}(t)\mathbf{q}_k$. This is very good for a DSP implementation where the available amount of fast memory is limited. As a comparison note that NLMS requires $\mathcal{O}(n)$ values to be stored and RLS requires $\mathcal{O}(n^2)$.

For a real-time DSP implementation of a AEC/SAEC algorithm to be feasible, the cost of processing each sample should not be too large. CG-RLS is easily found to be of the computational complexity $\mathcal{O}((n+M)\log_2(n+M))$ which is typically much lower than the computational complexity of RLS which is $\mathcal{O}(n^2)$. It is, however, larger than the computational complexity of NLMS which is $\mathcal{O}(n)$ and unfortunately the proportionality constant is quite large (and is in fact also directly proportional to the number of iterations performed per sample in the CG algorithm). Thus, for the CG-RLS algorithm to be computationally feasible in real-time echo cancellers, only one iteration in the CG algorithm should be performed per sample. As we will see the adaptation of CG-RLS appears to be sufficiently fast anyway.

The complexity comparison is, however, not entirely fair since CG-RLS is highly parallellizable as many of the heavy computations are performed using FFT/IFFT which can easily be computed in a parallel manner.

2.3.2 Application to SAEC

The application of the algorithm to MAEC is straightforward but for SAEC the matrix $\mathbf{X}(t)$ must be modified so that $\tilde{\mathbf{X}}(t) \triangleq [\mathbf{X}_1(t)\ \mathbf{X}_2(t)]$, where $\mathbf{X}_1(t)$ is defined as

$$\mathbf{X}_1(t) \triangleq \begin{bmatrix} \overline{x_1(t-M+1)} & \dots & \dots & \overline{x_1(t-M-n+2)} \\ \vdots & \ddots & & \vdots \\ \overline{x_1(t-n+1)} & & \ddots & \vdots \\ \vdots & \ddots & & \overline{x_1(t-M+1)} \\ \vdots & & \ddots & \vdots \\ \overline{x_1(t)} & \dots & \dots & \overline{x_1(t-n+1)} \end{bmatrix} \tag{2.4}$$

and $\mathbf{X}_2(t)$ is defined similarly. Note that the product $\mathbf{R}(t)\mathbf{q}_k \triangleq \tilde{\mathbf{X}}^*(t)\tilde{\mathbf{X}}(t)\mathbf{q}_k$ can still be performed efficiently using FFT/IFFT. The estimated filter $\mathbf{h}(t)$ then becomes $\mathbf{h}(t) \triangleq [\mathbf{h}_{11}(t)\ \mathbf{h}_{12}(t)]^T$.

2.3.3 Controlling the misalignment

The misalignment is usually not a problem for MAEC (assuming that all the necessary conditions for the input signal are fulfilled) since then (2.1) has a unique solution $\mathbf{h}(t)$ as only one input signal ($x_1(t)$) is used. Thus for RLS and NLMS the misalignment is not a problem in the MAEC case. In CG-RLS, however, the length M of the data window is finite (compared to RLS that has an infinite data window length)[2]. Furthermore, to keep the computational complexity as low as possible, M should be as small as possible (as a minimum we have $M = n$ since otherwise $\mathbf{R}(t)$ is not full rank and (2.1) will not have a unique solution). Using a short data window to solve (2.3) will minimize the error in (2.1), but since a short data window is used the estimate of $\mathbf{h}(t)$ will be extremely noisy and the misalignment will be large. The numerical simulations indicate that this might not be a problem if only one (or just a few) iteration of the CG algorithm is performed per sample since the dependence in the CG algorithm on the initial estimate then is so large that the solution will depend much on data from earlier data-windows. Thereby the effect of the noise is much lower than if (2.3) would be solved exactly using many iterations in the CG-RLS algorithm per sample

[2] If a forgetting factor is used the effective length of the data window is shorter.

and the misalignment is kept low. This effect can be seen as an averaging, or low-pass filtering.

For SAEC, (2.3) does in general almost not have a unique solution $\mathbf{h}(t)$ and thus the problem with misalignment using CG-RLS still remains even though we use only one iteration in the CG algorithm per sample. NLMS and RLS also suffer from this problem. Therefore a half-wave rectifier [2] was tested to decorrelate the signals $x_1(t)$ and $x_2(t)$ and was found to reduce the misalignment substantially. The half-wave rectifier introduces a nonlinearity in the input signals by setting

$$x_1'(t) = x_1(t) + \gamma \frac{x_1(n) + |x_1(n)|}{2}, \tag{2.5}$$

$$x_2'(t) = x_2(t) + \gamma \frac{x_2(n) - |x_2(n)|}{2}, \tag{2.6}$$

where γ is a user parameter. This approach is of course only feasible if it retains the stereophonic and acoustic properties of the sound signals. However, in [2] this is claimed to be the case for $\gamma \leq 0.5$.

2.4 Numerical examples

To demonstrate the performance of CG-RLS for MAEC and SAEC we have performed numerical simulations, comparing CG-RLS to NLMS and RLS. For NLMS the step size parameter was set to the standard choice of $\mu = 0.7$ to get a good adaptation speed while not being too sensitive to noise. (To get the fastest adaptation $\mu = 1$ should be chosen, but since this increases the sensitivity of the NLMS algorithm to noise a lower value is usually chosen).

For the RLS algorithm the standard initialization of the P matrix, $P(t_0) = \rho I$, was used where ρ was initially set to $\rho = 100$ which is large compared to the magnitudes of the signals $x_1(n)$ and $x_2(n)$. Thus the initial effect of ρ will disappear quickly [14] and the dependence on the initial value will be minor, as would be desired. No forgetting factor was used for RLS. In practice a forgetting factor must be used in RLS to accommodate for variations in the echo path, and since no forgetting factor is used the results for the RLS algorithm are somewhat better than they would be in practice.

To minimize the complexity of CG-RLS the length of the data window was chosen as $M = n$ for the mono case and $M = 2n$ for the stereo case which is the minimum to make $\mathbf{R}(t)$ full rank (which is a required for (2.1) to have a unique solution). Note that this is the worst choice from an estimation point of view since the estimates will be noisy due to the short data window. In order to minimize the computational complexity one iteration of the CG algorithms was performed per sample.

Note that CG-RLS requires $M+n$ data samples for MAEC and $2n+M$ for SAEC to start the algorithm while NLMS and RLS only require n samples. This is easily seen in the figures for the numerical results and should be taken into account when comparing the performance of the algorithms.

2.4.1 Performance measures

To assess the performance of the algorithms two measures were used: the Echo Return Loss Enhancement (ERLE) which measures the echo cancellation performance, and the misalignment which measures the accuracy of the impulse response estimation. The ERLE is defined as

$$ERLE = -10\log_{10}\left(\frac{\sigma_e^2}{\sigma_d^2}\right) \Rightarrow ERLE(t) \approx -10\log_{10}\left(\frac{\sum_{t-L+1}^{t} e^2(t)}{\sum_{t-L+1}^{t} d^2(t)}\right), \quad (2.7)$$

where L is the length of a sliding window for the ERLE estimation, and the misalignment is defined as

$$M(t) = \frac{||\mathbf{h}(t) - \tilde{\mathbf{h}}||^2}{||\tilde{\mathbf{h}}||^2}, \quad (2.8)$$

where $\tilde{\mathbf{h}}$ is the true room impulse response truncated to be of the same length as $\mathbf{h}(t)$. (For SAEC the misalignment is computed separately for $\mathbf{h_{11}(t)}$ and $\mathbf{h_{12}(t)}$). For an algorithm to be well-suited for MAEC/SAEC the ERLE value should be as high as possible and the misalignment should be as low as possible.

2.4.2 MAEC simulations

MAEC was tested using a 1 second long speech sequence sampled at 16 kHz as a far-end signal and impulse responses obtained by a high-resolution acoustic impulse response prediction code [6] (based on the theory presented in [1]). Four data sets were generated according to the described model of hands-free telephony in the monophonic case using four different time-invariant impulse responses of 600 filter taps each as the true room impulse responses. For each data set the signal to noise ratio (SNR) was adjusted to 40 dB (roughly corresponding to a hands-free telephony setup in a room with a low noise level) by adding white Gaussian noise (WGN) of suitable variance to the microphone input signal. To cancel the echo NLMS, RLS and CG-RLS were applied to all data sets to compute room impulse response estimates of lengths $n = 500$ filter taps each (shorter than the true room impulse responses since these are in reality of infinite length and thus always longer than the estimated impulse responses).

The results are displayed in Figures 2.2 and 2.3 where the misalignment and the ERLE achieved by the algorithms for the four different data sets are plotted. As can be seen in Figure 2.2 the misalignment is very low for CG-

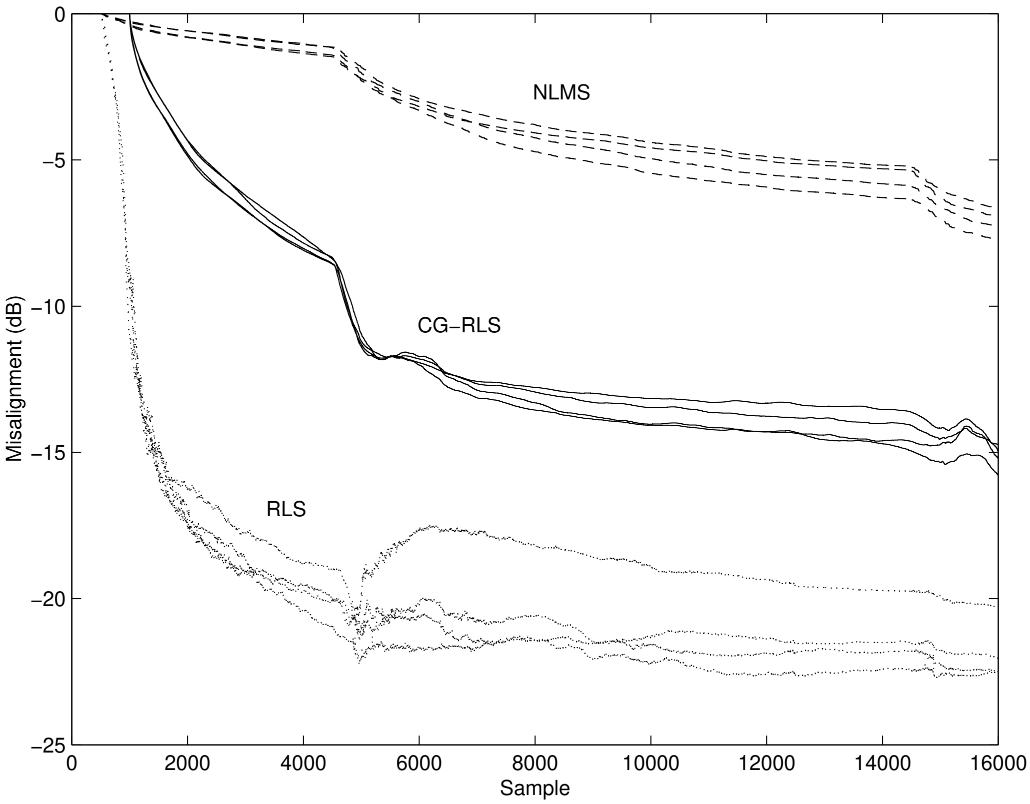

Figure 2.2: Misalignment for MAEC for the four different impulse responses using RLS (dotted), CG-RLS (solid) and NLMS (dashed).

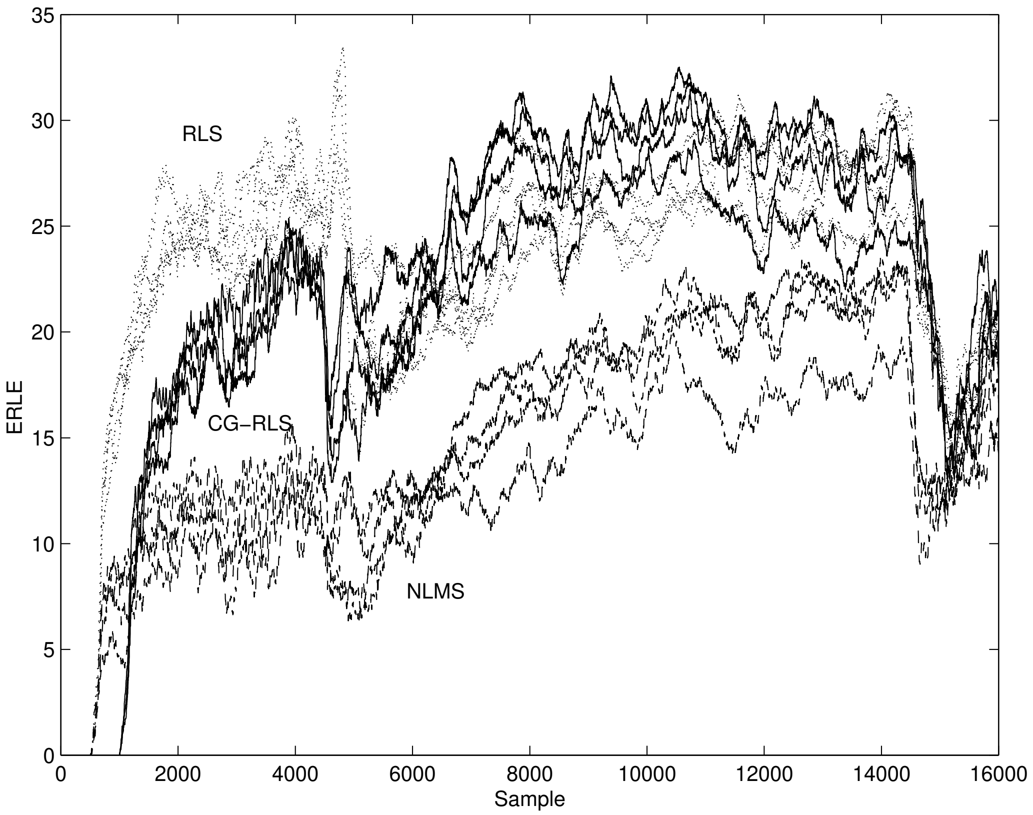

Figure 2.3: Echo cancellation performance for MAEC for the four different impulse responses using RLS (dotted), CG-RLS (solid) and NLMS (dashed).

RLS which indicates that there are no problems with the short data window when one iteration in the CG algorithm is performed per sample. Furthermore, from the plot of the misalignment it is apparent that the adaptation of RLS is much faster than that of CG-RLS, but also that the adaptation of CG-RLS is much faster than that of NLMS.

The echo cancellation performance is shown in Figure 2.3 where the values of the ERLE for the different algorithms applied to the different data sets are plotted as a function of time. From this figure it is clear that NLMS performs worst in terms of ERLE. CG-RLS performs similarly to RLS.

At first sight it could seem strange that for some samples CG-RLS even outperforms RLS in terms of ERLE since both algorithms minimize (2.1). One should, however, keep in mind that the data window for RLS is infinite while it is finite (and relatively short) for CG-RLS and therefore the minimum of (2.1) achieved by CG-RLS can be smaller than that for RLS. The filter estimates obtained by CG-RLS will, however, be much more noisy which is easily seen in the plot of the misalignment where the misalignment for RLS is much smaller than that for CG-RLS.

Another striking feature of the plots of Figure 2.3 is the "valleys" in the plots of the ERLE values. It might seem strange that the echo cancellation performance suddenly gets much worse at some points although the misalignment does not. This is, however, caused by the influence of the unmodeled tails of the true room impulse responses (remember that the estimated impulse responses are always shorter than the true impulse responses). When the power of the input signal (which consists of speech) abruptly decreases the influence of the unmodeled tail on the echo cancellation performance will be much larger than otherwise and there will be a reduction in the echo cancellation performance.

2.4.3 SAEC simulations

As in the MAEC simulations a 1 second long speech sequence sampled at 16 kHz was used as a far-end signal. Eight data sets, four with half-wave rectification and four without, were generated according to the described model of hands-free telephony in the stereophonic case using four different sets of four time-invariant impulse responses each (obtained by a high-resolution acoustic impulse response prediction code [6]) as true room impulse responses. The true impulse responses for the near-end room ($\mathbf{h_{11}}$ and $\mathbf{h_{12}}$) were 300 filter taps in length and the true impulse responses for the far-end room ($\mathbf{g_1}$ and $\mathbf{g_2}$) were 250 filter taps in length[3]. For each data set the SNR was adjusted to 40 dB by adding white Gaussian noise (WGN) of suitable variance to $y(t)$. For the sets where half-wave rectification was applied the parameter α in (2.5) and (2.6) was set to $\alpha = 0.5$ to provide a high decorrelation of the inputs signals $x_1(t)$ and $x_2(t)$ while not affecting the audio quality significantly. Note that the half-wave rectification yields signals with a somewhat lower SNR than that of the original signal. To cancel the echo NLMS, RLS

[3]The true room impulse responses used for the SAEC simulations were kept shorter than those used for the MAEC simulations since the available computational power was not sufficient for performing SAEC using RLS with the dimensions of the impulse responses used in the MAEC simulations.

and CG-RLS were used on all data sets to compute estimates of the near-end room impulse responses of lengths $n = 250$ filter taps each.

The results obtained without a half-wave rectifier are displayed in Figures 2.4 and 2.5 where the misalignment and ERLE achieved by the different algorithms and data sets are plotted. As can be seen in Figure 2.4, where the

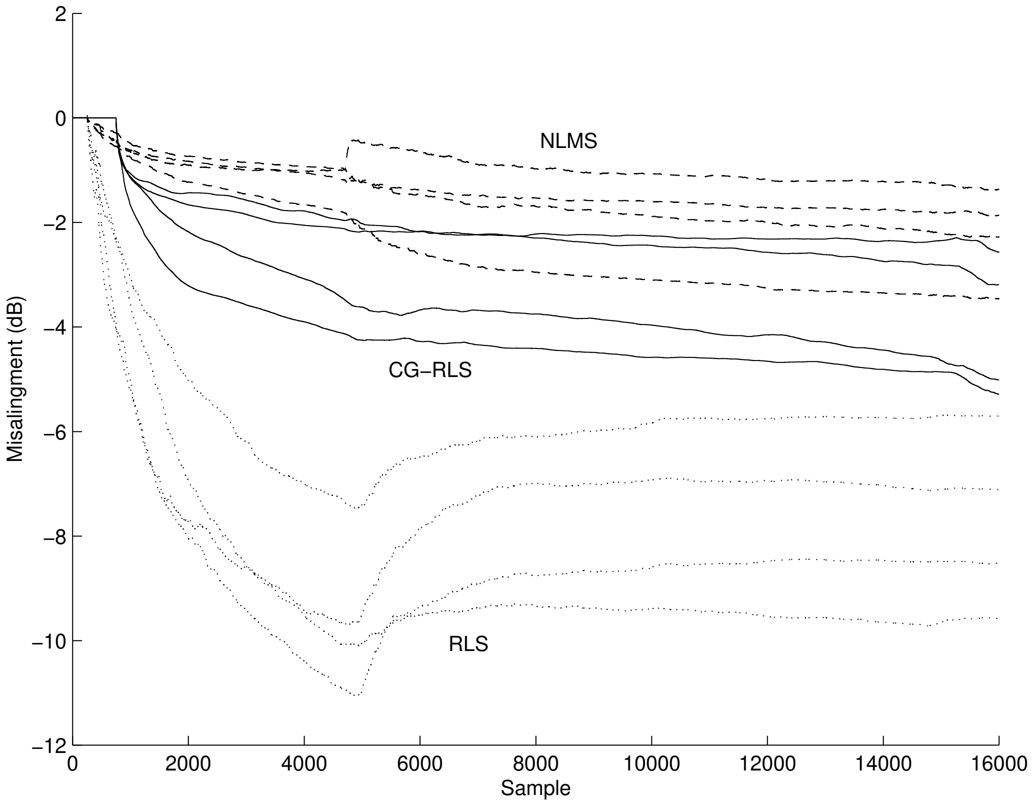

Figure 2.4: Misalignment for SAEC without half-wave rectification for the four sets of impulse responses using RLS (dotted), CG-RLS (solid) and NLMS (dashed).

misalignments for the estimates of $\mathbf{h_{11}}(t)$ are plotted (the misalignments for the estimates of $\mathbf{h_{12}}(t)$ behave similarly), the misalignment is very high for NLMS and CG-RLS and relatively high for RLS. From Figure 2.5 it is clear that the echo cancellation performance is good for RLS and CG-RLS with RLS performing best and NLMS performing relatively poorly. However, as

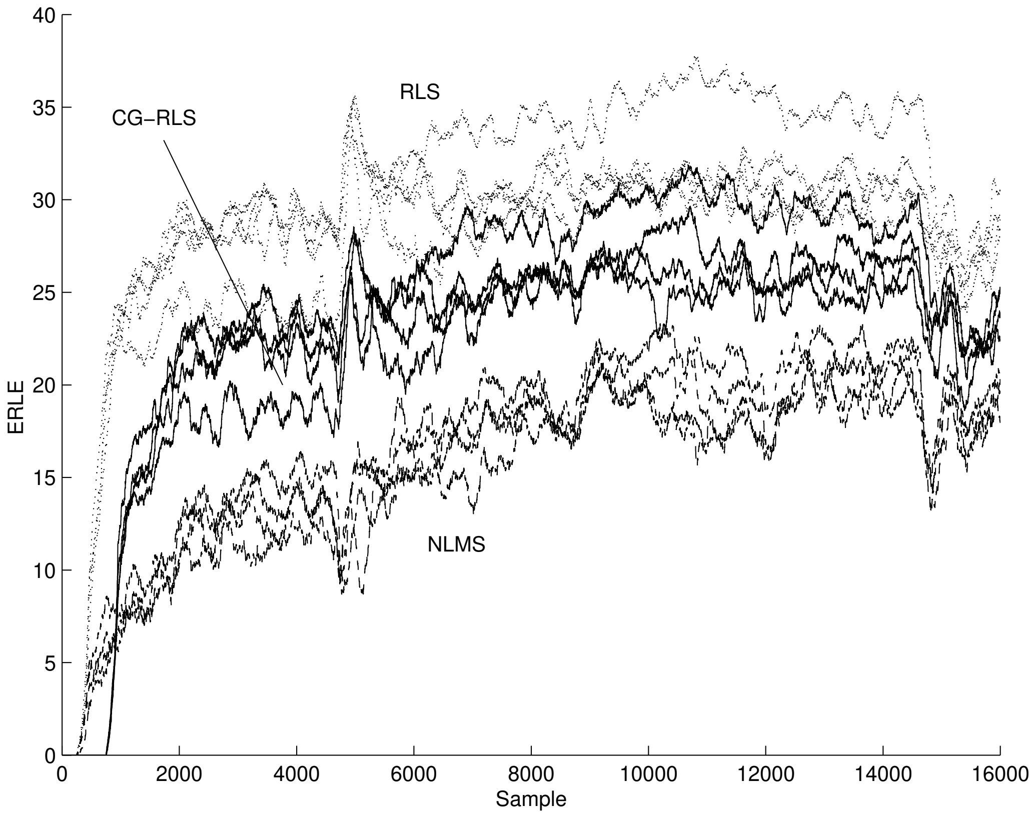

Figure 2.5: Echo cancellation performance for SAEC without half-wave rectification for the four sets of impulse responses using RLS (dotted), CG-RLS (solid) and NLMS (dashed).

has been noted before, the most important thing for echo cancellation is that the misalignment is small, and the misalignment is in this case unacceptably high, especially for NLMS and CG-RLS.

The results obtained with a half-wave rectifier are displayed in Figures 2.6 and 2.7 where the misalignment and ERLE achieved by the algorithms for the different data sets are plotted. As can be seen in Figure 2.6 the misalignments for the estimates obtained by all algorithms are significantly lower than those obtained when not using a half-wave rectifier. It is clear that RLS performs best but CG-RLS is not much worse. Both are significantly better than NLMS. From Figure 2.7 we can see that CG-RLS and RLS has a

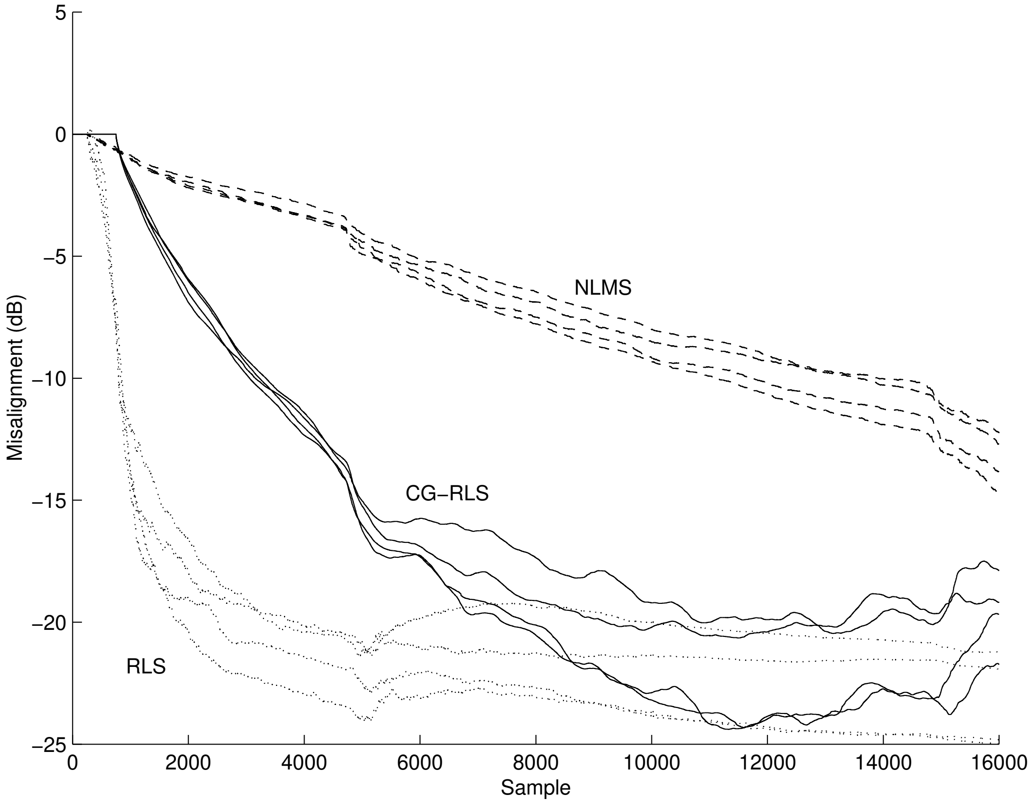

Figure 2.6: Misalignment for SAEC simulations with half-wave rectification using RLS (dotted), CG-RLS (solid) and NLMS (dashed).

comparable echo cancellation performance and that NLMS performs worst.

2.5 Concluding remarks

In acoustic echo cancellation a fast adaptation of the filter estimates to the true room impulse response(s) is important. Perhaps even more important is, however, that the computational complexity and storage requirements for the acoustic echo canceller are kept low to make real-time implementation feasible. An algorithm that appears to fulfill these requirements is the CG-RLS algorithm that was introduced in [11], and applied to the MAEC and SAEC problems in this paper. The numerical results show that it performs

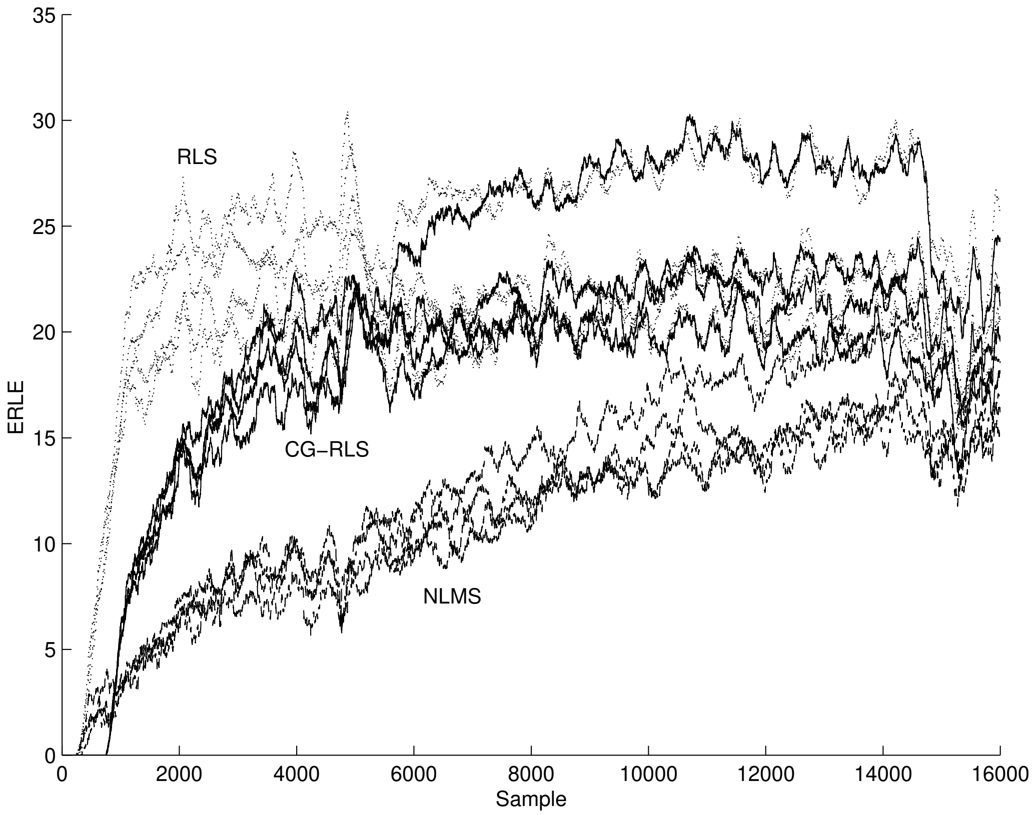

Figure 2.7: Echo cancellation performance for SAEC simulations with half-wave rectification using RLS (dotted), CG-RLS (solid) and NLMS (dashed).

much better than NLMS, which is one of the standard algorithms for AEC today. It performs, however, worse than RLS but as is shown the computational complexity and the storage requirements are much lower than those for RLS, which makes a real-time implementation for AEC feasible. This is hardly the case for RLS. CG-RLS is, however, much more computationally complex than NLMS but if that increase in computational complexity is affordable, CG-RLS could be the preferred choice.

Bibliography

[1] Allen JB, Berkley DA. Image method for efficiently simulating small-room acoustics. *Journal of the Acoustical Society of America*, 65:943–950, 1979.

[2] Benesty J, Morgan DR, Sondhi MM. A better understanding and an improved solution to the specific problems of stereophonic acoustic echo cancellation. *IEEE Transactions on Speech and Audio Processing*, 6(2):156–165, Mar 1998.

[3] Breining C, Dreiseitel P, Hänsler E, Mader A, Nitsch B, Puder H, Schertler T, Schmidt G, Tilp J. Acoustic Echo Control - An Application of Very-High-Order Adaptive Filters. *IEEE Signal Processing Magazine*, 16(4):42–69, July 1999.

[4] Dreiseitel P, Hänsler E, Puder H. Acostic echo and noise control - a long lasting challenge. In *Proc. EUSIPCO-98, 9th European Signal Processing Conference*, Sept 1998.

[5] Eneroth P, Gänsler T, Gay S, Benesty J. Studies of a Wideband Stereophonic Acoustic Echo Canceler. In *Proc. 1999 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, New York*, Oct 1999.

[6] Garas J. Room Impulse Response v2.3 software and documentation, Dec 2000. http://www.dspalgorithms.com/download.html.

[7] Golub G, Van Loan CF. *Matrix Computations.* The John Hopkins University Press, 3rd edition, 1996.

[8] Hänsler E. The hands-free problem - An annotated bibliography. *Signal Processing*, 3(27):259–271, 1992.

[9] Haykin S. *Adaptive Filter Theory*. Prentice-Hall, Upper Saddle River, New Jersey, 3rd edition, 1996.

[10] Ljung L, Söderström T. *Theory and Practice of Recursive Identification*. The MIT Press, 1983.

[11] Ng MK, Plemmons RJ. Fast Recursive Least Squares Adaptive Filtering by Fast Fourier Transform-based Conjugate Gradient Iterations. *SIAM Journal of Scientific Computing*, 17(4):920–941, July 1996.

[12] Ozeki K, Umeda T. An Adaptive Filtering Algorithm Using an Orthogonal Projection to an Affine Subspace and Its Properties. *Electronics and Communications in Japan*, 67-A(5):19–27, 1984.

[13] Sondhi MM. An adaptive echo canceller. *Bell Syst. Tech. J.*, 46:497–510, Mar 1967.

[14] Stoica P, Åhgren P. Exact initialization of the recursive least-squares algorithm. *Preprint*, 2000.

Chapter 3

On the hierarchical RLS algorithm

On the hierarchical RLS algorithm

Abstract

A so-called hierarchical recursive least-squares (HRLS) algorithm was suggested in a recent letter in an attempt to reduce the computational burden and improve the convergence rate of the classical RLS algorithm. The discussion of HRLS in the original letter, however, has many unclear points; in particular no clear explanation was offered for the good simulation results reported. In this letter we provide some analysis of the HRLS to determine when this algorithm may be expected to work or fail. It turns out that the input to the channel must be a white sequence, otherwise HRLS may yield grossly biased estimates of the channel FIR coefficients.

3.1 Introduction and preliminaries

Consider a discrete-time FIR channel model given by

$$y(t) = h_0 u(t) + \cdots + h_{n-1} u(t-n+1) + e(t), \quad t = 1, 2, \ldots, \tag{3.1}$$

where $u(t)$ is the input to the channel, $y(t)$ is the output, $e(t)$ is a noise term, $\{h_k\}$ are the FIR channel coefficients, and $n < \infty$ is the channel order. Using the definitions

$$\phi(t) = [u(t) \ldots u(t-n+1)]^*, \tag{3.2}$$

$$\theta = [h_0 \ldots h_{n-1}]^T, \tag{3.3}$$

we can rewrite (3.1) in the following more compact form:

$$y(t) = \phi^*(t)\theta + e(t). \tag{3.4}$$

The problem of interest is the estimation of θ from a collection of observations $\{y(t), \phi(t)\}_{t=1}^N$. In what follows we consider the least squares (LS) estimate of θ (see, e.g., [1]) and the hierarchical LS (HLS) algorithm of [3]. For the sake of clarity we focus on the batch versions of these algorithms instead of the recursive variants considered in [3]. As RLS is nothing but a sequential (either exact or approximate, depending on the initialization used, see e.g [2]) implementation of the batch LS, the discussion that follows applies almost verbatim to the RLS and HRLS algorithms as well. We also note that we concentrate the present discussion on the model in (3.1) because this appears to be the one considered in [3]. Even so, the conclusions of our study apply mutatis mutandis to any other linear regression models of the general form in (3.4).

3.2 LS and subspace-constrained LS

The *LS estimate* of θ minimizes the quadratic criterion

$$\sum_{t=1}^{N} |y(t) - \phi^*(t)\theta|^2 \tag{3.5}$$

and is given by

$$\hat{\theta}_{LS} = R^{-1}r, \tag{3.6}$$

where

$$R = \sum_{t=1}^{N} \phi(t)\phi^*(t)\,;\; r = \sum_{t=1}^{N} \phi(t)y(t) \tag{3.7}$$

(see,e.g., [1]). In some cases we may possess the apriori information that the true parameter vector θ belongs (or is very close to) the range space of a given matrix U of dimensions $n \times \bar{n}$ (with $n > \bar{n}$). This means that θ can be written as

$$\theta = U\alpha \tag{3.8}$$

for some (unknown) $\bar{n} \times 1$-vector α. The *subspace-constrained LS estimate* of θ is obtained by minimizing (3.5) under the constraint (3.8). Equivalently, we can insert (3.8) into (3.5), minimize the so-obtained criterion w.r.t. α, and then use the so obtained α in (3.8), which results in the following readily verified estimate of θ:

$$\hat{\theta}_{CLS} = U(U^*RU)^{-1}U^*r. \tag{3.9}$$

It is easy to prove that, under (3.8), $\hat{\theta}_{CLS}$ is statistically more accurate than $\hat{\theta}_{LS}$, as expected (we leave the proof of this fact to the reader).

The above discussion on the subspace-constrained LS estimate will prove useful in understanding the HLS algorithm. As we show in the next section, the latter algorithm provides an estimate of θ that is a special case of (3.9) corresponding to a certain U matrix.

3.3 HLS

The HLS can be implemented in many ways, depending on how we factor the order n (see [3] for details). Here, for convenience's sake, we focus on a simple implementation. Let n be a perfect square, define $m = n^{1/2}$, and split the $\phi(t)$ vector in m subvectors of size $m \times 1$ as follows:

$$\phi(t) = \begin{bmatrix} \phi_1^T(t) & \cdots & \phi_m^T(t) \end{bmatrix}^T . \tag{3.10}$$

Define, similarly to (3.6), (3.7)

$$R_k = \sum_{t=1}^{N} \phi_k(t)\phi_k^*(t)\,;\; r_k = \sum_{t=1}^{N} \phi_k(t)y(t) \tag{3.11}$$

and the LS estimates in the "first level" of HLS:

$$\hat{\beta}_k = R_k^{-1} r_k, \quad (k = 1, \ldots, m). \tag{3.12}$$

The "second (and last, in the case of $m = n^{1/2}$) level" of the HLS consists in defining new "regression variables"

$$\mu_k(t) = \phi_k^*(t)\hat{\beta}_k, \quad (k = 1, \ldots, m), \tag{3.13}$$

$$\mu(t) = [\mu_1(t) \ldots \mu_m(t)]^*, \tag{3.14}$$

and a new LS estimate obtained by regressing $y(t)$ onto $\mu(t)$:

$$\hat{\gamma} = \left[\sum_{t=1}^{N} \mu(t)\mu^*(t) \right]^{-1} \sum_{t=1}^{N} \mu(t)y(t). \tag{3.15}$$

Finally, $\hat{\gamma}$ is used to obtain the following estimate of the noise-free output in (3.1):

$$\hat{y}(t) = \mu^*(t)\hat{\gamma}. \tag{3.16}$$

Compared with the intricate and rather unclear discussion of the H(R)LS in [3], the above explanation of HLS makes it possible to put the final result of the algorithm in a clear-cut form. To this end, note that

$$\mu(t) = \begin{bmatrix} \hat{\beta}_1^* \phi_1(t) \\ \vdots \\ \hat{\beta}_m^* \phi_m(t) \end{bmatrix} = \begin{bmatrix} \hat{\beta}_1^* & & 0 \\ & \ddots & \\ 0 & & \hat{\beta}_m^* \end{bmatrix} \phi(t) \triangleq U^* \phi(t), \tag{3.17}$$

where

$$U = \begin{bmatrix} \hat{\beta}_1 & & 0 \\ & \ddots & \\ 0 & & \hat{\beta}_m \end{bmatrix} \quad (n \times m). \tag{3.18}$$

Inserting (3.17) in (3.15) and the resulting expression for $\hat{\gamma}$ in (3.16) yields:

$$\hat{y}(t) = \phi^*(t) U (U^* R U)^{-1} U^* r. \tag{3.19}$$

As this is an estimate of $\phi^*(t)\theta$, it follows that the HLS estimate of θ is given by:

$$\hat{\theta}_{HLS}(t) = U (U^* R U)^{-1} U^* r, \tag{3.20}$$

with U given by (3.18).

Hence we proved that, as indicated in the previous section, *$\hat{\theta}_{HLS}$ is a subspace-constrained LS estimate corresponding to a (data-dependent) matrix U given by (3.18).* The conclusion is that $\hat{\theta}_{HLS}$ may be a more accurate estimate of θ than $\hat{\theta}_{LS}$ provided that θ belongs or more realistically is "close enough" to the range space of U in (3.18). Quantitative statements about the closeness of θ to the range space of (3.18) for $\hat{\theta}_{HLS}$ to possibly outperform $\hat{\theta}_{LS}$ may be attempted, but we will not dwell on such aspects here. Instead we focus on the main point we want to make, which can be explained as follows.

Let

$$\theta = \begin{bmatrix} \theta_1^T & \cdots & \theta_m^T \end{bmatrix}^T, \tag{3.21}$$

where $\{\theta_k\}$ are $m \times 1$-vectors. Owing to the structure of U in (3.18), it is clear that θ is close to $range(U)$ if and only if θ_k is close to $\hat{\beta}_k$ (or a scaled version thereof). *If the vectors* $\{\phi_k(t)\}$ *are uncorrelated with each other*, then this is true (at least for $N >> 1$) since $\hat{\beta}_k \to \theta_k$ in such a case (as $N \to \infty$). Note that in the case of (3.1) the previous condition on $\{\phi_k(t)\}$ translates into the requirement that $\{u(t)\}$ *is a white sequence* (which may be reasonable to assume in communication applications). Also note that in the simulations in [3] several subvectors $\{\theta_k\}$ of θ were equal to zero. As explained above HLS may be expected to work well only if the estimates $\{\hat{\beta}_k\}$ computed in the first step of the algorithm are such that the differences $|\theta_k - c_k\hat{\beta}_k|$ can be made "small enough" (for $k = 1, \ldots, m$) by choosing $\{c_k\}$. If $\theta_k = 0$ then we can simply satisfy this condition for an *arbitrary* $\hat{\beta}_k$ by choosing $c_k = 0$. Hence, parameter vectors θ with many zero subvectors $\{\theta_k\}$ constitute a favorable case for HLS. In summary, if $\{u(t)\}$ in (3.1) is a colored signal then in general the HLS algorithm cannot be expected to work well and hence should not be used (unless "many" subvectors $\{\theta_k\}$ of θ are known to be (almost) zero, see below). If $\{u(t)\}$ is white then HLS may work reasonably well but not necessarily better than LS; if, in addition, several subvectors $\{\theta_k\}$ of θ are equal or close to zero (i.e., the channel has only a few nonzero (multipath) coefficients or the channel order is overestimated), then HLS may be expected to outperform LS.

3.4 Numerical examples

To illustrate the previous results we consider the following examples. In all of them, $n = 16$ (hence $m = 4$), $e(t)$ is a white noise sequence with zero mean and standard deviation 0.4 and $u(t)$ is generated as a first order autoregressive sequence,

$$u(t) = au(t-1) + w(t) \tag{3.22}$$

where $w(t)$ is a white sequence with zero mean and variance equal to unity. To estimate θ we use $\hat{\theta}_{LS}$, (3.6), and $\hat{\theta}_{HLS}$, (3.18)(3.20). In each case we run 200 Monte-Carlo simulations corresponding to different input and noise sequences, compute the empirical mean squared errors $\{MSE_k\}_{k=1}^{n}$ for each element of the parameter vector, and plot $\frac{1}{n}\sum_{k=1}^{n} MSE_k \triangleq MSE$ (in dB).

First, let $a = 0$, hence $u(t)$ is a white sequence, and let $h_0 = h_1 = h_2 = h_3 = 1$ and all other elements in θ be zero. Figure 3.1 shows the average

MSE as a function of N. As predicted by our analysis, in this case $\hat{\theta}_{HLS}$

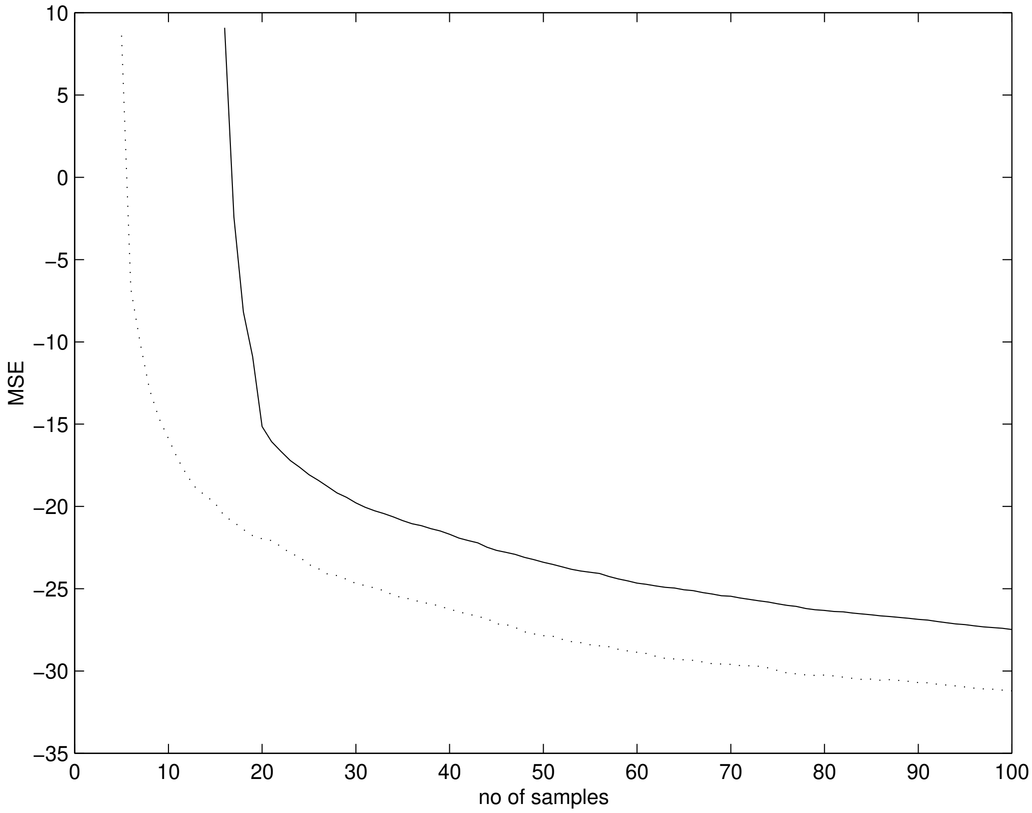

Figure 3.1: MSE in the estimation of $\theta = [1\,1\,1\,1\,0\,\ldots\,0]^T$ (n=16) using HLS (dotted line) and LS (solid line) for uncorrelated input, as a function of the number of samples used for the estimation.

performs better than $\hat{\theta}_{LS}$ for all values of N considered. Up to $N = 30$, HLS needs about 10 samples less than LS to achieve the same MSE, and this gain increases with N: LS with $N = 110$ has approximately the same MSE as HLS with $N = 60$. Note that this is no contradiction to the fact that $\hat{\theta}_{LS}$ is the best *unbiased* estimate of θ, since $\hat{\theta}_{HLS}$ is typically a *biased* estimate (a biased estimate may be better than the best unbiased one, in terms of MSE, by slightly compromising the bias for a significant decrease in variance).

Next, we let θ be as in the previous example and a be 0.99 to study the effect of a correlated input sequence. From the results it was clear that HLS

outperformed LS in this case as well, which is to be expected since we had only one subvector θ_1 that was different from zero and thus the correlation of the input signal did not affect the results.

We repeated the two examples above by increasing the number of nonzero elements in θ to 8 and 16 (so that we had more nonzero subvectors θ_k of θ). As in the previous examples, the nonzero elements of θ were set equal to one. It was clear from the results that HLS performed worse than LS. Moreover the difference between the performances of LS and HLS increased as the number of non-zero elements increased.

HLS breaks down (it yields highly biased estimates) when the input sequence is correlated. Figure 3.2 shows the MSE of the estimates of θ for this case as a function of N with 8 and 16 non zero elements in θ. As before the correlated input sequence is generated by setting $a = 0.99$ in (3.22).

3.5 Conclusions

Let us assume that $\{u(t)\}$ is a white sequence, as this is a weak requirement in many communication applications. If, in addition, we expect that the (multipath) channel has only a few non-zero coefficients, then we should use H(R)LS instead of (R)LS for channel estimation. On the other hand, if we suspect that most of the channel coefficients are rather different from zero, then we can use H(R)LS only for the first $2n$ samples after which we may switch to using (R)LS.

To conclude, note that biased estimation is not an easy topic. Finding a biased estimate that outperforms the (best unbiased) LS estimate even in special cases is a difficult task. This fact motivated our interest in HLS, which as we showed in this letter can outperform LS (albeit perhaps in too special a case).

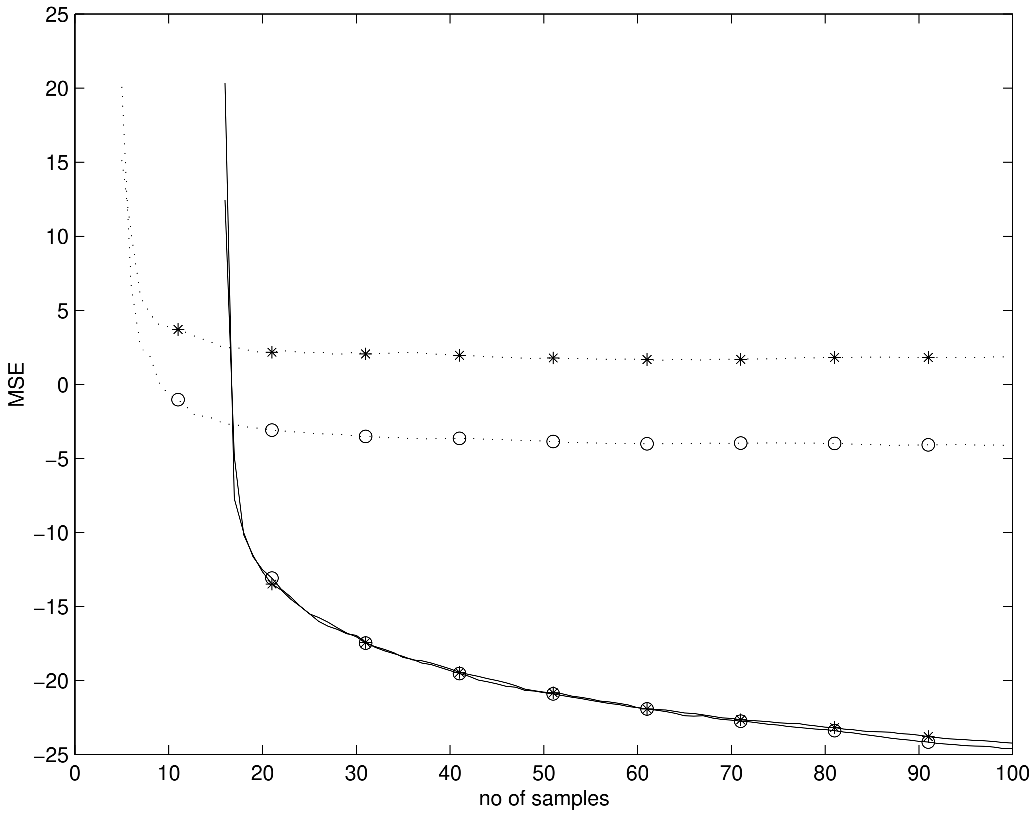

Figure 3.2: MSE in the estimation of $\theta = [1\,1\,1\,1\,1\,1\,1\,1\,0\,\ldots\,0]^T$ (circles) and of $\theta = [1\,\ldots\,1]^T$ (stars) (n=16) using HLS (dotted line) and LS (solid line) for correlated input, as a function of the number of samples used for the estimation.

Bibliography

[1] Söderström T, Stoica P. *System Identification.* Prentice-Hall, International, UK, 1989.

[2] Åhgren P Stoica P. Exact Initialization of the Recursive Least-Squares Algorithm. *International Journal of Adaptive Control and Signal Processing*, 2001, to appear.

[3] Woo TK. HRLS : A More Efficient RLS Algorithm for Adaptive FIR Filtering. *IEEE Communications Letters*, 5:81–84, March 2001.

Chapter 4

Exact initialization of the recursive least-squares algorithm

Exact initialization of the recursive least-squares algorithm

Summary

We present an initialization procedure for the recursive least-squares (RLS) algorithm that has almost the same form as the RLS algorithm itself and which is exact in the sense that the so-initialized RLS estimate coincides with the batch LS estimate as soon as the latter exists.

4.1 Introduction and problem formulation

Consider the following linear regression model:

$$y(t) = \varphi^*(t)\theta + e(t), \qquad t = 1, \dots, N, \tag{4.1}$$

where N denotes the number of available data samples: $\{y(t), \varphi(t)\}_{t=1}^{N}$; $e(t)$ is a noise term; θ is the vector of unknown parameters with $\dim(\theta) \triangleq n$; and the superscript "$*$" stands for the conjugate transpose (or just the transpose for real-valued data). As is well known (see, e.g., [5][8][4]) the batch LS estimate of θ is given by

$$\hat{\theta}_N = R_N^{-1} r_N, \tag{4.2}$$

where the index N is meant to show that the corresponding quantity was obtained from N data samples, and

$$r_N = \sum_{t=1}^{N} \varphi(t) y(t), \tag{4.3}$$

$$R_N = \sum_{t=1}^{N} \varphi(t) \varphi^*(t). \tag{4.4}$$

As is also well known the recursive LS algorithm (RLSA) that updates the LS estimate sequentially in N consists of the following equations ([5][8][4][1]):

$$\hat{\theta}_N = \hat{\theta}_{N-1} + K_N \varepsilon(N), \tag{4.5a}$$

$$\varepsilon(N) = y(N) - \varphi^*(N)\hat{\theta}_{N-1}, \tag{4.5b}$$

$$K_N = \frac{P_{N-1}\varphi(N)}{1 + \varphi^*(N)P_{N-1}\varphi(N)}, \tag{4.5c}$$

$$P_N = P_{N-1} - K_N \varphi^*(N) P_{N-1}, \tag{4.5d}$$

where $P_N = R_N^{-1}$ (this inverse is assumed to exist, hence the RLS algorithm in (4.5) is not applicable for $N < n$). Note that the RLSA can be re-written in various forms some of which may have better numerical properties than the form above (we refer to the cited references for details on this aspect that falls beyond the scope of the present paper).

The RLSA must be provided with initial values for $\hat{\theta}_N$ and P_N. *One possibility* to obtain these initial values consists of accumulating m pairs of observations $\{y(t), \varphi(t)\}_{t=1}^m$ and using

$$\hat{\theta}_m = P_m r_m, \qquad P_m = R_m^{-1} \tag{4.6}$$

to initialize the RLSA (m must be such that R_m^{-1} exists). This initialization is exact (in the sense that the RLSA initialized by (4.6) will compute the batch LS estimate for any $N > m$, modulo numerical round-off errors), yet it has been rarely used probably because of the following drawbacks:

(i) the need to store $\{y(t), \varphi(t)\}_{t=1}^m$;

(ii) the need for a computer code that is rather different from that of the RLSA (in particular, (4.6) would require a matrix inversion routine); and

(iii) the comparatively large computational effort needed to obtain (4.6), which is not distributed in time but concentrated at $t = m$.

The above drawbacks are of particular importance for low-cost hardware applications, such as channel equalization. The *other well-known possibility* to initialize the RLSA consists of setting

$$\hat{\theta}_1 = 0 \qquad \text{and} \qquad P_1 = cI, \tag{4.7}$$

where $c > 0$ is a "suitable chosen" constant ([5][8]). This initialization scheme is by far the most commonly-used one probably because it has none of the drawbacks (i)-(iii) above. However it has its own problems:

(iv) it is not exact in the sense that the RLSA started-up using (4.7) provides an estimate that coincides with the batch LS estimate only asymptotically (for $N \to \infty$); and

(v) clear-cut guidelines for choosing the user parameter c are not readily available.

The two problems (iv) and (v) above are related to one another in that a solution to (v) would solve (iv) as well. Indeed the estimate computed by the RLSA initialized via (4.7) can be written in a batch manner as ([5][8]):

$$\hat{\theta}_N = [R_N + (1/c)I]^{-1} r_N. \tag{4.8}$$

The regularized LS estimate above may have a smaller mean square error than the LS estimate in (4.2) provided that c is "appropriately chosen" (to compromise bias for variance). In such a case the fact that the initialization in (4.7) is not exact (see (iv) above) would be turned into an advantage. However choosing c in such a way is hardly possible without knowledge about the true parameter values, and hence the initialization in (4.7) usually suffers from both drawbacks (iv) and (v) above. Note that for a "large" c, the convergence of the RLSA (initialized as in (4.7)) to the exact LS estimate will be rapid, but the variability of the estimated parameters in the transitory phase may be significant. Conversely, for a "small" c the initial variability of the estimates will be kept small, but the convergence to the exact LS estimates may be rather slow (see [5] [8] and [3] for details on this aspect).

Our goal in this paper is to present an initialization scheme for the RLSA that is *exact* and *user parameter-free* (hence it does not suffer from the problems (iv) and (v) associated with (4.7)), and which additionally has *almost the same form* as the RLSA itself (and hence does not have the drawbacks (i)-(iii) of (4.6) either).

Our main working assumption is the following:

A: The vectors $\{\varphi(1), \ldots, \varphi(n)\}$ (where $n = \dim(\theta)$) are linearly independent.

This condition, which is satisfied in most cases, allows us to avoid a number of technicalities in developing the exact initialization scheme. In particular, under **A** we do not have to choose any subjective threshold required on a finite precision-machine to decide whether, for example, $\varphi(n)$ is in the linear span of $\{\varphi(1), \ldots, \varphi(n-1)\}$. The reader interested in the general case where $\{\varphi(1), \varphi(2), \text{etc}\}$ may be linearly dependent should consult [1] (see also the unpublished technical report [7] where the main result in [1] was rederived in an independent manner). The excellent early paper [1] on LS parameter estimation contains an exhaustive treatment of the RLSA including an exact initialization scheme that is a generalization of ours. However, the approach in [1] is much more intricate than ours, which may be one of the reasons why the results of [1] are not well known in the system identification and signal processing communities. We hope that the more direct approach of the present paper, which should be easy to follow, will promote the use of the exact initialization scheme introduced in [1] (under general conditions) and re-derived here in a simple manner (under the fairly weak condition **A** above).

We should note that we obtained the exact initialization scheme of this paper before becoming aware of [1], which explains in part why our approach and that in [1] are fairly different from one another. Our attention to the problem of exactly initializing the RLSA was triggered by [6], which (unaware of [1]) has also attempted to solve this problem but, in our opinion, has fallen short of providing a satisfactory solution (the solution in [6] requires the direct computation of the pseudoinverse of R_m (for $m = 1, \ldots, n$), and hence it is not attractive from a computational standpoint).

Finally, we remark on the fact that (as we already mentioned) the RLSA can be rewritten in several other forms of which the QR decomposition(QRD)-based variant is probably best known. The exact initialization of the QRD-based RLSA is discussed, for example, in ([2], Chap.14) for the so-called "pre-windowed" form of the regressor vectors $\{\varphi(t)\}$.

4.2 Exact initialization scheme

As already stated, there is an implicit assumption in (4.2) (and also in (4.5)) that the inverse R_N^{-1} exists. Whenever this is not the case we write (4.2) in the form of the so-called "normal matrix equation":

$$R_N \hat{\theta}_N = r_N. \tag{4.9}$$

Let

$$\Phi_N = [\varphi(1) \dots \varphi(N)], \qquad y_N = \begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix}, \tag{4.10}$$

and rewrite (4.9) as

$$\Phi_N \Phi_N^* \hat{\theta}_N = \Phi_N y_N. \tag{4.11}$$

For all facts stated in what follows we assume that condition **A** holds true.

Fact 1. *For $N < n$ the system of linear equations in (4.9) has an infinite number of solutions (note that a general system of the form of (4.9) with $det(R_N) = 0$ may not have any solution).*

Proof. For $N < n$ and under **A**, Φ_N has full column rank equal to N and hence (4.9) or (4.11) is equivalent to

$$\Phi_N^* \hat{\theta}_N = y_N \tag{4.12}$$

whose set of solutions is given by

$$\hat{\theta}_N = \Phi_N (\Phi_N^* \Phi_N)^{-1} y_N + \mu_N, \tag{4.13}$$

where μ_N is any vector in the null space of Φ_N^*. □

Of all possible solutions of (4.9) (for $N < n$) we choose the minimum-norm solution (see, e.g., [8][1][6]):

$$\hat{\theta}_N = R_N^{\dagger} r_N, \tag{4.14}$$

where $R_N^\dagger$ denotes the Moore-Penrose pseudoinverse of R_N. Note that for $N < n$ and under **A** (e.g. [8]):

$$R_N^\dagger = \Phi_N(\Phi_N^*\Phi_N)^{-2}\Phi_N^*. \tag{4.15}$$

The *basic idea* of the exact initialization scheme is to update $\hat{\theta}_N$ in (4.14) and $P_N = R_N^\dagger$ for $N = 1, 2, \ldots, n$, and then use the so-obtained $\hat{\theta}_n$ and P_n (which coincide with the corresponding RLS "state variables" computed in a batch manner) as initial values for the RLSA.

First we derive an updating formula for

$$P_N = R_N^\dagger, \qquad \text{for } N = 1, 2, \ldots, n. \tag{4.16}$$

Let, for $N \leq n$,

$$\Pi_N = R_N^\dagger R_N = R_N R_N^\dagger = \Phi_N(\Phi_N^*\Phi_N)^{-1}\Phi_N^* \tag{4.17}$$

denote the orthogonal projector onto the range space of Φ_N, $\mathcal{R}(\Phi_N)$, and let

$$\breve{\Pi}_N = I - \Pi_N \tag{4.18}$$

denote the orthogonal projector onto the null space of Φ_N^*, $\mathcal{N}(\Phi_N^*)$. Also, let

$$K_N = \frac{\breve{\Pi}_{N-1}\varphi(N)}{\varphi^*(N)\breve{\Pi}_{N-1}\varphi(N)}. \tag{4.19}$$

Note that for $N \leq n$ the denominator in (4.19) is different from zero (it is strictly positive) under **A**.

Fact 2. *For $N = 2, \ldots, n$,*

$$P_N = P_{N-1} - K_N\varphi^*(N)P_{N-1} - P_{N-1}\varphi(N)K_N^* + \tag{4.20}$$

$$+ [1 + \varphi^*(N)P_{N-1}\varphi(N)]K_N K_N^*, \tag{4.21}$$

with

$$P_1 = \frac{\varphi(1)\varphi^*(1)}{[\varphi^*(1)\varphi(1)]^2}. \tag{4.22}$$

Proof. By a standard formula for the inverse of a partitioned matrix (see, e.g., [8]) we have that

$$(\Phi_N^*\Phi_N)^{-1} = \left(\begin{bmatrix}\Phi_{N-1}^* \\ \varphi^*(N)\end{bmatrix}\begin{bmatrix}\Phi_{N-1} & \varphi(N)\end{bmatrix}\right)^{-1} = \begin{bmatrix} I \\ 0\end{bmatrix}(\Phi_{N-1}^*\Phi_{N-1})^{-1}\begin{bmatrix} I & 0\end{bmatrix} +$$
$$+ \begin{bmatrix}-(\Phi_{N-1}^*\Phi_{N-1})^{-1}\Phi_{N-1}^*\varphi(N) \\ 1\end{bmatrix}\begin{bmatrix}-\varphi^*(N)\Phi_{N-1}(\Phi_{N-1}^*\Phi_{N-1})^{-1} & 1\end{bmatrix} /$$
$$/[\varphi^*(N)\varphi(N) - \varphi^*(N)\Phi_{N-1}(\Phi_{N-1}^*\Phi_{N-1})^{-1}\Phi_{N-1}^*\varphi(N)]. \tag{4.23}$$

This implies that

$$(\Phi_N^*\Phi_N)^{-1}\Phi_N^* = (\Phi_N^*\Phi_N)^{-1}\begin{bmatrix}\Phi_{N-1}^* \\ \varphi^*(N)\end{bmatrix} =$$
$$= \begin{bmatrix}(\Phi_{N-1}^*\Phi_{N-1})^{-1}\Phi_{N-1}^* - \dfrac{(\Phi_{N-1}^*\Phi_{N-1})^{-1}\Phi_{N-1}^*\varphi(N)\varphi^*(N)\breve{\Pi}_{N-1}}{\varphi^*(N)\breve{\Pi}_{N-1}\varphi(N)} \\ \dfrac{\varphi^*(N)\breve{\Pi}_{N-1}}{\varphi^*(N)\breve{\Pi}_{N-1}\varphi(N)}\end{bmatrix}. \tag{4.24}$$

Hence,

$$\begin{aligned} R_N^\dagger &= \Phi_N(\Phi_N^*\Phi_N)^{-1}(\Phi_N^*\Phi_N)^{-1}\Phi_N^* = \\ &= R_{N-1}^\dagger - R_{N-1}^\dagger\varphi(N)K_N^* - K_N\varphi^*(N)R_{N-1}^\dagger + \\ &+ K_N\varphi^*(N)R_{N-1}^\dagger\varphi(N)K_N^* + K_NK_N^*, \end{aligned} \tag{4.25}$$

which proves (4.21). As (4.22) follows from the definition of P_1, the proof is concluded. ☐

Next we derive an updating formula for $\hat{\theta}_N$ (for $N = 2, \ldots, n$). A simple calculation shows that:

$$\hat{\theta}_N = R_N^\dagger r_N = R_N^\dagger(r_N - R_N\hat{\theta}_{N-1} + R_N\hat{\theta}_{N-1}) =$$
$$= \Pi_N\hat{\theta}_{N-1} + R_N^\dagger[r_{N-1} + \varphi(N)y(N) - R_{N-1}\hat{\theta}_{N-1} - \varphi(N)\varphi^*(N)\hat{\theta}_{N-1}]. \tag{4.26}$$

However, $R_{N-1}\hat{\theta}_{N-1} = r_{N-1}$ by the definition of $\hat{\theta}_{N-1}$. Furthermore, as $\hat{\theta}_{N-1} \in \mathcal{R}(\Phi_{N-1})$ (see, e.g (4.14) and (4.15)) and $\mathcal{R}(\Phi_{N-1}) \subset \mathcal{R}(\Phi_N)$, it also follows that $\Pi_N\hat{\theta}_{N-1} = \hat{\theta}_{N-1}$. Hence, (4.26) reduces to:

$$\hat{\theta}_N = \hat{\theta}_{N-1} + R_N^\dagger\varphi(N)\varepsilon(N), \tag{4.27}$$

where

$$\varepsilon(N) = y(N) - \varphi^*(N)\hat{\theta}_{N-1}. \tag{4.28}$$

Note that

$$\hat{\theta}_1 = \frac{\varphi(1)y(1)}{\varphi^*(1)\varphi(1)}. \tag{4.29}$$

Fact 3. *For $N = 2, \dots, n$,*

$$R_N^\dagger \varphi(N) = K_N, \tag{4.30}$$

where K_N is as defined in (4.19).

Proof. Postmultiplying (4.25) by $\varphi(N)$ yields at once:

$$\begin{aligned} R_N^\dagger \varphi(N) = R_{N-1}^\dagger \varphi(N) - R_{N-1}^\dagger \varphi(N) - K_N[\varphi^*(N)R_{N-1}^\dagger \varphi(N)] + \\ + K_N[\varphi^*(N)R_{N-1}^\dagger \varphi(N)] + K_N = K_N. \end{aligned} \tag{4.31}$$

□

Finally we need an updating equation for the matrix $\breve{\Pi}_N$ that appears in the formula for K_N (see (4.19)).

Fact 4. *For $N = 2, \dots, n$,*

$$\breve{\Pi}_N = \breve{\Pi}_{N-1} - \frac{\breve{\Pi}_{N-1}\varphi(N)\varphi^*(N)\breve{\Pi}_{N-1}}{\varphi^*(N)\breve{\Pi}_{N-1}\varphi(N)} = \breve{\Pi}_{N-1} - K_N\varphi^*(N)\breve{\Pi}_{N-1}, \tag{4.32}$$

with

$$\breve{\Pi}_1 = I - \frac{\varphi(1)\varphi^*(1)}{\varphi^*(1)\varphi(1)}. \tag{4.33}$$

Proof. It follows from (4.24) that

$$\begin{aligned} \Pi_N = \begin{bmatrix} \Phi_{N-1} & \varphi(N) \end{bmatrix} (\Phi_N^*\Phi_N)^{-1}\Phi_N^* = \Pi_{N-1} - \frac{\Pi_{N-1}\varphi(N)\varphi^*(N)\breve{\Pi}_{N-1}}{\varphi^*(N)\breve{\Pi}_{N-1}\varphi(N)} + \\ + \frac{\varphi(N)\varphi^*(N)\breve{\Pi}_{N-1}}{\varphi^*(N)\breve{\Pi}_{N-1}\varphi(N)} = \Pi_{N-1} + \frac{\breve{\Pi}_{N-1}\varphi(N)\varphi^*(N)\breve{\Pi}_{N-1}}{\varphi^*(N)\breve{\Pi}_{N-1}\varphi(N)}, \end{aligned} \tag{4.34}$$

which proves (4.32). Because (4.33) follows from the definition of $\breve{\Pi}_1$, the proof is finished. □

The derivation of the exact initialization scheme for the RLSA is now complete: the scheme involves three "state variables" $\hat{\theta}_N$, $\breve{\Pi}_N$ and P_N (one in excess of the RLSA) and consists of the equations (4.27) for updating $\hat{\theta}_N$, (4.32) for updating $\breve{\Pi}_N$ and (4.21) for updating P_N. For reader's convenience we summarize these equations along with those of the RLSA in Table 4.1. In order to present this summary in a unitary manner we re-denote $\breve{\Pi}_N$ by P_N for $N = 1, 2, \ldots, n$ (since the updating equation for $\breve{\Pi}_N$ for $N = 2, \ldots, n$ and that for P_N for $N > n$ have an identical form and, moreover, these two variables play a similar role in the updating of $\hat{\theta}_N$ for $N = 2, \ldots, n$ and, respectively, $N = n + 1, \ldots$); accordingly, we re-denote P_N for $N = 1, \ldots, n$ by S_N.

As can be observed from Table 4.1 the exact initialization scheme proposed herein has the same implementation form as the RLSA, with the exception of a slight difference in the formula for K_N and the need for an extra updating equation (for S_N).

4.3 Numerical examples

To evaluate the exact initialization of the RLSA, we consider the following linear regression

$$y(t) = \begin{bmatrix}\sin(\omega_a t) & \sin(\omega_b t)\end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} + e(t) \triangleq \varphi^*(t)\theta + e(t), \tag{4.35}$$

where $e(t)$ is a white Gaussian noise with zero mean and unit variance, ω_a and ω_b are given, and $a = b = \sqrt{20}$ (resulting in a SNR of 10 dB per signal component) are the parameters to be estimated.

Two cases will be considered: a well-conditioned case when $|\omega_a - \omega_b|$ is relatively large: $\omega_a = 0.5\frac{\pi}{2}$, $\omega_b = 0.2\frac{\pi}{2}$; and an ill-conditioned case when $|\omega_a - \omega_b|$ is quite small: $\omega_a = 0.5\frac{\pi}{2}$, $\omega_b = 0.49\frac{\pi}{2}$. For each case 10 independent realizations of $\{e(t)\}$ were generated and the RLSA was used to estimate the parameters a and b. Apart from the exact initialization of the RLSA, the initialization (4.7) was also used with two different values for c: one "small" ($c = 1$) and one "large" ($c = 100$).

The results for the well-conditioned case are displayed in Figures 4.1-4.3 where the estimates of a are plotted for all 10 realizations (the estimates of b behave similarly). From these figures it can be seen that the estimates

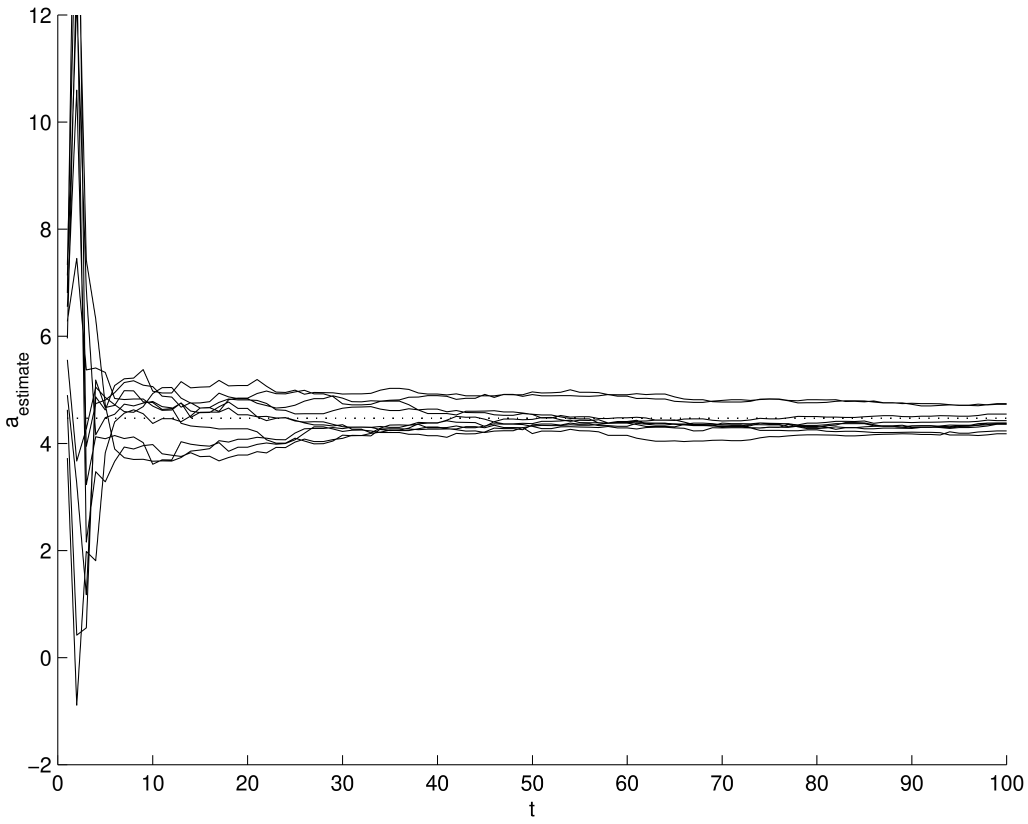

Figure 4.1: Estimates of a using the RLSA with exact initialization (the true parameter value is given by the dashed line).

using the exact initialization are similar to the estimates using the initialization (4.7) with a "large" c, while the estimates using the initialization (4.7) with a "small" c are worse. This was expected since the estimation problem is well conditioned and the exact initialization of the RLSA results in the unbiased LS estimate while the initialization (4.7) only gives an asymptotically unbiased estimate. In this case the best initialization to use should thus be the exact one.

The results for the ill-conditioned case are displayed in Figures 4.4-4.6. As can be seen the ill-conditioning causes the estimates using the exact initialization and the estimates using the initialization (4.7) with a "large" c to be rather noisy, while the estimates obtained by the RLSA initialized by

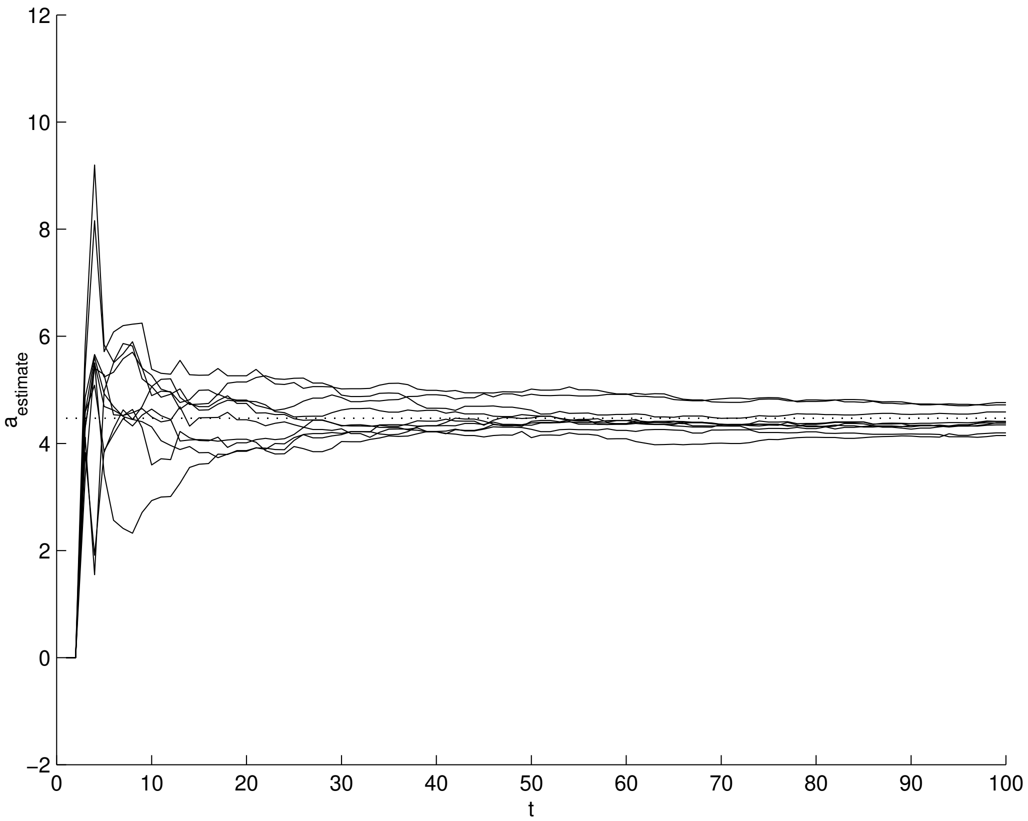

Figure 4.2: Estimates of a using the RLSA initialized by (4.7) with $c = 100$ (the true parameter value is given by the dashed line).

(4.7) with a "small" c are better due to the regularizing effect of $(1/c)I$ in (4.8).

Among other things these results demonstrate the problem of choosing the "right" value for c in the initialization (4.7). Without knowing anything about the true parameter values the choice of c is not an easy task. For instance in the ill-conditioned case above we may well choose a "too large" value for c and hence miss the beneficial effects of regularization.

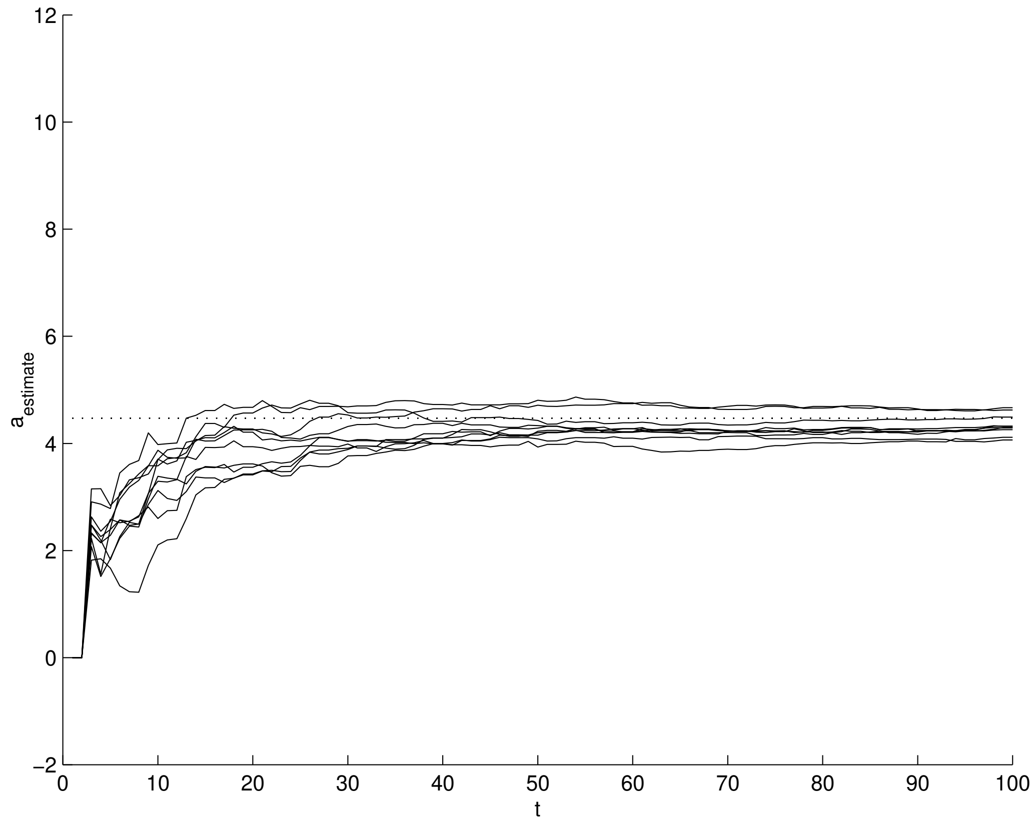

Figure 4.3: Estimates of a using the RLSA initialized by (4.7) with $c = 1$ (the true parameter value is given by the dashed line).

4.4 Concluding remarks

The exact initialization scheme for the RLSA introduced in [1] and re-derived in a simple manner in this paper (under a fairly weak condition) has an obvious theoretical importance for the classical theory of RLS estimation. Additionally, the scheme appears to be usually more practically useful than its main competitor (see (4.7)) that may outperform it (in a mean square sense) only if a certain user parameter is “carefully chosen”.

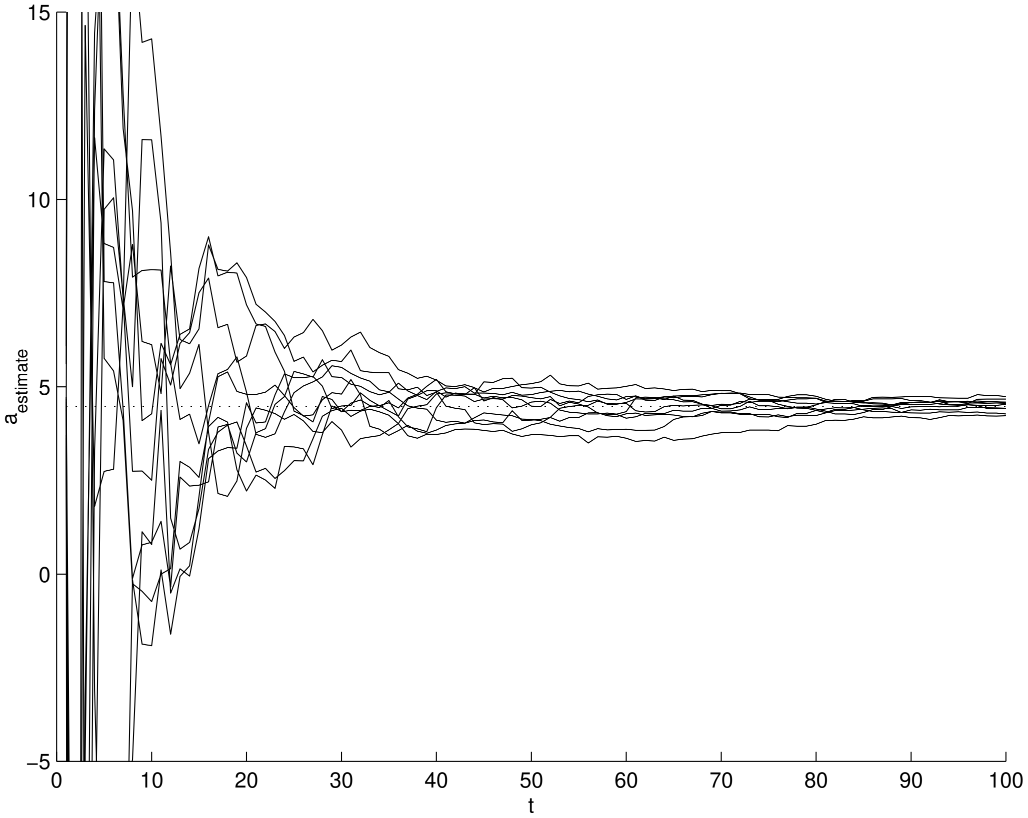

Figure 4.4: Estimates of a using the RLSA with exact initialization (the true parameter value is given by the dashed line).

- **For** $N = 1:$

$$\left|\begin{array}{l} \hat{\theta}_1 = \dfrac{\varphi(1)y(1)}{\varphi^*(1)\varphi(1)} \\ P_1 = I - \dfrac{\varphi(1)\varphi^*(1)}{\varphi^*(1)\varphi(1)} \\ S_1 = \dfrac{\varphi(1)\varphi^*(1)}{[\varphi^*(1)\varphi(1)]^2} \end{array}\right.$$

- **For** $N = 2, \ldots, n:$

$$\left|\begin{array}{l} \text{(a) } \hat{\theta}_N = \hat{\theta}_{N-1} + K_N\varepsilon(N) \\ \text{(b) } \varepsilon(N) = y(n) - \varphi^*(N)\hat{\theta}_{N-1} \\ \text{(c) } K_N = \dfrac{P_{N-1}\varphi(N)}{\varphi^*(N)P_{N-1}\varphi(N)} \\ \text{(d) } P_N = P_{N-1} - K_N\varphi^*(N)P_{N-1} \\ \text{(e) } S_N = S_{N-1} - K_N\varphi^*(N)S_{N-1} - S_{N-1}\varphi(N)K_N^* + \\ \qquad\quad +[1 + \varphi^*(N)S_{N-1}\varphi(N)]K_NK_N^* \end{array}\right.$$

- **For** $N = n:$

$$\left|\begin{array}{l} \hat{\theta}_n = \hat{\theta}_n \\ P_n = S_n \end{array}\right.$$

- **For** $N = n+1, n+2, \ldots :$

$$\left|\begin{array}{l} (\tilde{a}) \text{ Same as (a) above} \\ (\tilde{b}) \text{ Same as (b) above} \\ (\tilde{c}) \; K_N = \dfrac{P_{N-1}\varphi(N)}{1 + \varphi^*(N)P_{N-1}\varphi(N)} \\ (\tilde{d}) \text{ Same as (d) above} \end{array}\right.$$

Table 4.1: Summary of the RLSA with the proposed exact initialization scheme.

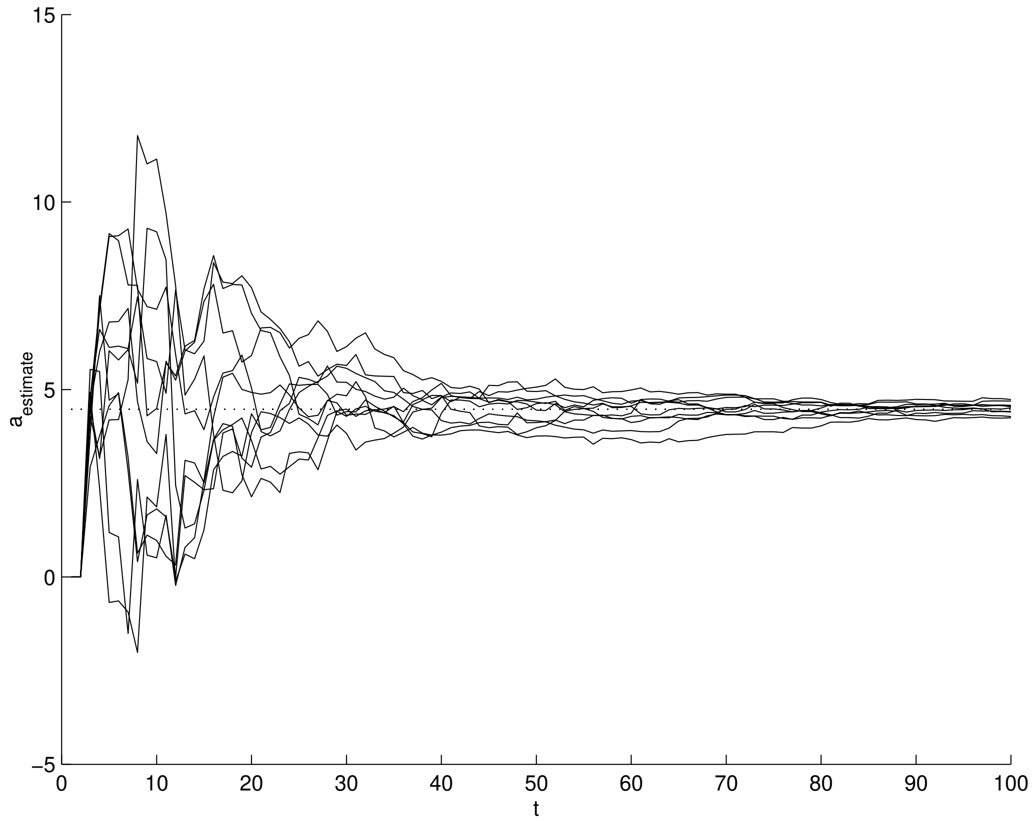

Figure 4.5: Estimates of a using the RLSA initialized by (4.7) with $c = 100$ (the true parameter value is given by the dashed line).

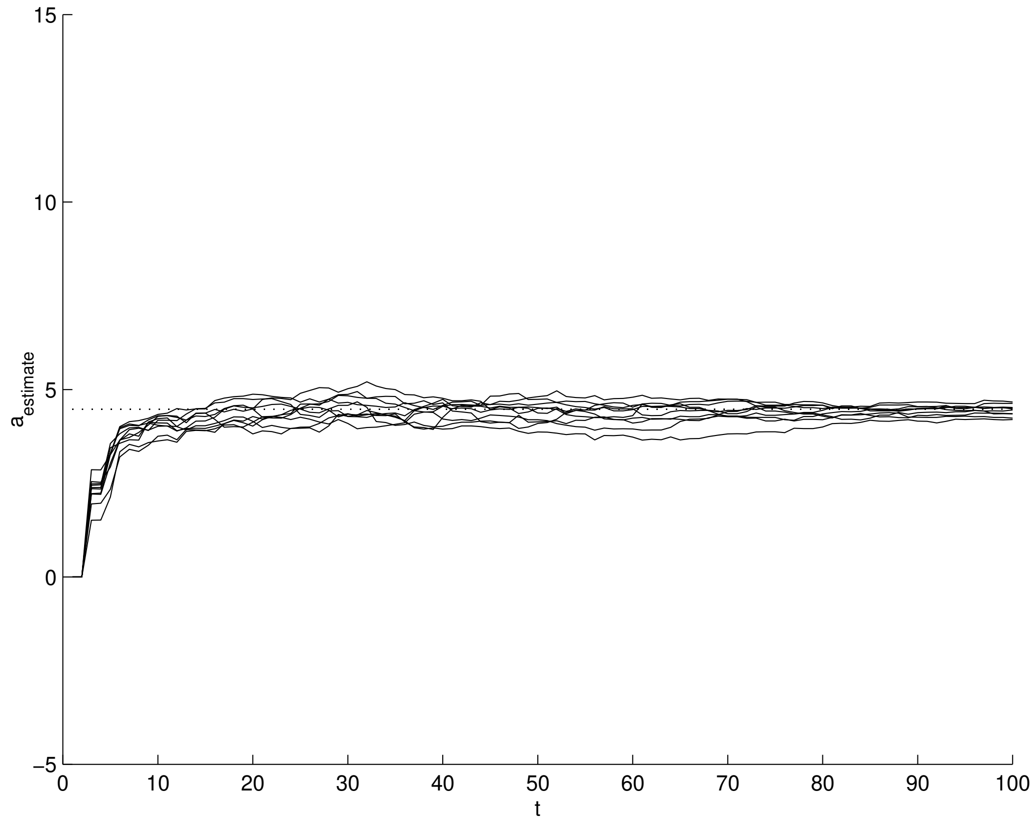

Figure 4.6: Estimates of a using the RLSA initialized by (4.7) with $c = 1$ (the true parameter value is given by the dashed line).

Bibliography

[1] Albert A, Sittler R. A method for computing least squares estimators that keep up with the data. *SIAM J. Control*, 3(3):384–417, 1965.

[2] Haykin S. *Adaptive Filter Theory.* Prentice-Hall, Upper Saddle River, New Jersey, 3rd edition, 1996.

[3] Hubing NE, Alexander ST. Statistical analysis of initialization methods for RLS adaptive filters. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 39:1793–1804, August 1991.

[4] Kailath T, Sayed AH, Hassibi B. *Linear Estimation.* Prentice-Hall, Upper Saddle River, New Jersey, 2000.

[5] Ljung L. *System Identification.* Prentice-Hall, Upper Saddle River, New Jersey, 2nd edition, 1999.

[6] Rong Li X. Initialization of recursive least squares algorithms with linear constraints. *Preprint*, 2000.

[7] Söderström T. Notes on pseudoinverses: Application to identification. Technical Report 7003, Lund Institute of Technology, July 1970.

[8] Söderström T, Stoica P. *System Identification.* Prentice-Hall, International, UK, 1989.

Chapter 5

Common Factor Estimation and Two Applications in Signal Processing

Common Factor Estimation and Two Applications in Signal Processing

Abstract

The problem of finding common factors (or common roots) of a set of polynomials without rooting is of interest in many fields of research. When the polynomials are observed in noise, i.e., their coefficients are corrupted by errors, the problem becomes challenging. In this paper we suggest a method for estimating the greatest common divisor (GCD) of a set of polynomials whose coefficients are perturbed by noise. The corresponding algorithm is called COFE (COmmon Factor Estimation). The COFE algorithm has several applications of which in this paper we discuss two in detail. One of these is MUSIC (MUltiple SIgnal Classification) which is reformulated as a COFE problem. The advantage of COFE MUSIC over the existing MUSIC is the ease by which we estimate the parameters. The other application is System Identification, where ML (Maximum Likelihood) estimates of the parameters of an ARARX system can be directly obtained by the suggested COFE algorithm in a simple manner.

5.1 Introduction

The problem of estimating the greatest common divisor (GCD) of a set of polynomials with perturbed coefficients appears in several signal processing applications, such as system identification by indirect predictor error methods [9][12], blind channel equalization based on second order statistics [4][15], symmetric non-causal auto-regressive modeling by subspace methods [13] and MUSIC for temporal and spatial frequency estimation [10][11].

A general treatment of the GCD estimation problem for the case of two noisy polynomials is presented in [12]. In most of the aforementioned applications, however, the number of polynomials dealt with is generally larger than two. In this paper we provide an algorithm which we call COFE (COmmon Factor Estimation) for the GCD estimation problem in the case of multiple polynomials. As the latter case appears to be significantly more complicated than the former, our treatment here is not as complete as that in [12] for the two polynomial case. In particular, we do not discuss the detection problem (i.e. the estimation of the GCD degree).

To illustrate the potential for applications of our suggested algorithm we focus on two signal processing problems: MUSIC for signal pole estimation and System Identification. The COFE based MUSIC algorithm is called COFE MUSIC. COFE MUSIC is shown to have important advantages over the existing MUSIC algorithms. In particular, unlike the latter, COFE MUSIC directly estimates the "signal poles" and hence it does not have any problem with separating the "signal poles" from the "extraneous (or noise) poles". Furthermore, COFE MUSIC can exploit the information that certain poles are multiple which other MUSIC algorithms cannot. COFE MUSIC is also able to deal with data containing multiple (more than one) damped sinusoids (with signal poles not on the unit circle). All these advantages come at no performance loss: the performance of COFE MUSIC is comparable to that of ROOT-MUSIC (the preferred version of MUSIC for uniform and linear sampling).

In the case of system identification it has been shown in [9][12] that the problem of maximum likelihood (ML) estimation of the parameters of an ARARX system (a system of difference equations with exogenous input and autoregressive (AR) residuals) can be reformulated as a COFE problem. Here we present an extension of the basic algorithm in [9][12] to ARARX systems with multiple inputs. It turns out that it performs similarly to the standard prediction error method (PEM) while being computationally less expensive than PEM.

5.2 The COFE Problem

Let

$$B_k(z) = b_{k,0} + b_{k,1}z + \cdots + b_{k,n+\bar{n}}z^{n+\bar{n}}, \qquad (k = 1, 2, \cdots, m) \tag{5.1}$$

be a set of m polynomials that have a GCD $C(z)$ of degree n

$$B_k(z) = C(z)D_k(z), \qquad (k = 1, 2, \cdots, m). \tag{5.2}$$

In the above equation we have

$$C(z) = c_0 + c_1 z + \cdots + c_n z^n \tag{5.3}$$

and

$$D_k(z) = d_{k,0} + d_{k,1}z + \cdots + d_{k,\bar{n}}z^{\bar{n}}, \qquad (k = 1, 2, \cdots, m). \tag{5.4}$$

The polynomials $\{D_k(z)\}$ are coprime (i.e. they have no common factor). The assumption that all the polynomials $\{B_k(z)\}$ are of same degree, $n+\bar{n}$, is only made for notational convenience and does not impose any restriction (i.e., we do not require that $b_{k,n+\bar{n}} \neq 0$). For the sake of generality we assume that the coefficients of the polynomials introduced above are complex valued. Let μ be the vector made from the coefficients of the polynomials in (5.1)

$$\mu = \begin{bmatrix} b_{1,0} & \cdots & b_{1,n+\bar{n}} & \cdots & b_{m,0} & \cdots & b_{m,n+\bar{n}} \end{bmatrix}^T \tag{5.5}$$

and let c denote the coefficient vector associated with the GCD

$$c = \begin{bmatrix} c_0 & c_1 & \cdots & c_n \end{bmatrix}^T. \tag{5.6}$$

We want to estimate c and $\{D_k(z)\}$ from a noise corrupted version $\hat{\mu}$ of the vector μ in (5.5).

In this paper we do not address the "detection problem", i.e., the estimation of the GCD degree. Hence we assume that n is given (therefore $\bar{n}$ is given as well). A solution to the detection problem could be obtained by extending the χ^2-test in [12] to the present case of $m \geq 2$. However, to simplify the exposition, we concentrate on the problem of GCD coefficient estimation and leave the detection problem for future studies.

5.3 Solution to the COFE Problem

The previous informal statement of the COFE problem can be turned into a formal statement by using the criterion

$$\|\hat{\mu} - \mu\|^2 = (\hat{\mu} - \mu)^*(\hat{\mu} - \mu) \tag{5.7}$$

which depends on $\{c_k\}$ and $\{d_{k,j}\}$ via μ. The superscript * denotes the complex conjugate transpose. We want to determine the parameters that minimize (5.7). In MUSIC, $\{c_k\}$ are the parameters of interest while $\{d_{k,j}\}$ are nuisance parameters, whereas in System Identification we are also interested in $\{d_{k,j}\}$. To this end, observe that μ can be written as

$$\mu = Gc \tag{5.8}$$

where

$$G = \begin{bmatrix} G_1^T & G_2^T & \cdots & G_m^T \end{bmatrix}^T \tag{5.9}$$

and G_k is an $(n + \bar{n} + 1) \times (n + 1)$ matrix given by,

$$G_k = \begin{bmatrix} d_{k,0} & 0 & \cdots & 0 \\ d_{k,1} & d_{k,0} & \ddots & \vdots \\ \vdots & d_{k,1} & \ddots & 0 \\ d_{k,\bar{n}-1} & \ddots & \ddots & d_{k,0} \\ d_{k,\bar{n}} & d_{k,\bar{n}-1} & \ddots & d_{k,1} \\ 0 & d_{k,\bar{n}} & \ddots & \vdots \\ \vdots & \ddots & \ddots & d_{k,\bar{n}-1} \\ 0 & \dots & 0 & d_{k,\bar{n}} \end{bmatrix}. \tag{5.10}$$

To make the factorization in (5.8) unique, we set $d_{1,0} = 1$ (which is just one of many possible normalizations). Because of the products between the parameters $\{c_k\}$ and $\{d_{k,j}\}$ in (5.8), the criterion (5.7) is a quartic function of the unknowns and therefore an exact explicit minimization of (5.7) is hardly possible. Below we show how to obtain an approximate closed-form minimizer which is a good approximation of the exact minimizer of (5.7) in the small error region (i.e., for small values of $\|\hat{\mu} - \mu\|$). The criterion in (5.7), with $\{d_{k,j}\}$ given, can easily be minimized with respect to $\{c_k\}$. The minimizing c vector is given by

$$\hat{c} = (G^* G)^{-1} G^* \hat{\mu}. \tag{5.11}$$

Replacing c in (5.7) and (5.8) by $\hat{c}$ we get the concentrated criterion,

$$\hat{\mu}^* \Pi_G^\perp \hat{\mu}, \tag{5.12}$$

where $\Pi_G^\perp$ is the orthogonal projector operator onto the null space of G^* $(\mathcal{N}(G^*))$,

$$\Pi_G^\perp = I - \Pi_G; \qquad \Pi_G = G(G^* G)^{-1} G^*. \tag{5.13}$$

Note that G has full column rank and also, for later use, that the dimension of $\mathcal{N}(G^*)$ is

$$dim[\mathcal{N}(G^*)] = m\bar{n} + (m-1)(n+1). \tag{5.14}$$

To obtain a simple approximate minimizer for the function in (5.12) we need to derive a convenient basis for $\mathcal{N}(G^*)$ and to this end we define the following matrix,

$$S_k = \begin{bmatrix} d^*_{k,0} & \cdots & d^*_{k,\bar{n}} & & 0 \\ & \ddots & & \ddots & \\ 0 & & d^*_{k,0} & \cdots & d^*_{k,\bar{n}} \end{bmatrix} \tag{5.15}$$

with dimensions $(n+\bar{n}+1)\times(2\bar{n}+n+1)$. We note, in passing that the matrix G_k equals the $(n+\bar{n}+1)\times(n+1)$ top-left block of S^*_k. It can easily be verified (see, e.g, [3][8]) that

$$\begin{bmatrix} G^*_i & G^*_j \end{bmatrix} \begin{bmatrix} -S_j \\ S_i \end{bmatrix} = 0, \qquad \text{(for } i,j = 1,2,\cdots,m). \tag{5.16}$$

Let $S^{(2)}$ denote the matrix whose columns span the nullspace of G for $m=2$, and let $S^{(m)}$ denote the corresponding matrix for m polynomials. We have the following lemma:

Lemma 1. *Let*

$$S^{(2)} = \begin{bmatrix} -S_2 \\ S_1 \end{bmatrix} \tag{5.17}$$

and

$$S^{(m)} = \begin{bmatrix} & -S_m & 0 & \cdots & \cdots & 0 \\ & 0 & -S_m & \ddots & \ddots & \vdots \\ S^{(m-1)} & \vdots & 0 & \ddots & \ddots & \vdots \\ & \vdots & \vdots & \ddots & \ddots & 0 \\ & 0 & 0 & \cdots & 0 & -S_m \\ 0 & S_1 & S_2 & \cdots & \cdots & S_{m-1} \end{bmatrix}. \tag{5.18}$$

The size of $S^{(m)}$ is $\frac{(m-1)^2}{2}(2\bar{n}+n+1)\times m(n+\bar{n}+1)$. Then the range space of the matrix $S^{(m)}$ coincides with the null space of the matrix G^*:

$$\mathcal{R}(S^{(m)}) = \mathcal{N}(G^*). \tag{5.19}$$

Proof 1. *The above lemma is similar to Theorem 1 of [3]. Unlike the latter here we do not have any constraints on the number of polynomials and their orders. However, the proof of the lemma can be obtained by following the steps of [3]. To avoid repetition we do not include the proof here.* $\square$

Note that, from now on, for notational convenience, we write $S^{(m)}$ as S. By making use of (5.19), we can rewrite the criterion in (5.12) as

$$\hat{\mu}^* S (S^* S)^{\dagger} S^* \hat{\mu} \tag{5.20}$$

where $(\cdot)^{\dagger}$ denotes the Moore-Penrose pseudoinverse (or, in fact, any other pseudoinverse). Next, let us assume that the magnitude of the errors in $\hat{\mu}$, $\|\hat{\mu} - \mu\|$, is "small". A consequence of this assumption is that

$$S^* \hat{\mu} = S^*(\hat{\mu} - Gc) = S^*(\hat{\mu} - \mu) \tag{5.21}$$

is also "small". The orthogonality property $S^* \mu = 0$ means that *initial estimates* of $\{d_{k,j}\}$ can be obtained by minimizing the quadratic function:

$$\|S^* \hat{\mu}\|^2. \tag{5.22}$$

Using a Taylor series expansion technique it is straightforward to show that the replacement of $(S^* S)^{\dagger}$ in (5.20) by a consistent estimate does not affect the asymptotic properties of the minimizer of (5.20). (Henceforth, "asymptotic" and "consistent" refer to properties that hold in the limit as $\|\hat{\mu} - \mu\|$ tends to zero). The estimate of S and therefore $(S^* S)^{\dagger}$ is obtained from the initial estimates of $\{d_{k,j}\}$ obtained above and the *enhanced estimates* of $\{d_{k,j}\}$ can be obtained by

$$\{\hat{d}_{k,j}\} = \underset{\{d_{k,j}\}}{arg\,min}\, \hat{\mu}^* S \; (\hat{V}_1 \hat{\Sigma}_1^{-2} \hat{V}_1^*) \; S^* \hat{\mu} \tag{5.23}$$

where $\hat{\Sigma}_1$ and $\hat{V}_1$ are defined by the singular value decomposition (SVD) of $\hat{S}$ (a consistent estimates of S) as

$$\hat{S} = \begin{bmatrix} \hat{U}_1 & \hat{U}_2 \end{bmatrix} \begin{bmatrix} \hat{\Sigma}_1 & 0 \\ 0 & \hat{\Sigma}_2 \end{bmatrix} \begin{bmatrix} \hat{V}_1^* \\ \hat{V}_2^* \end{bmatrix}. \tag{5.24}$$

Here $\hat{U}_1$, $\hat{U}_2$, $\hat{V}_1$ and $\hat{V}_2$ are (semi)unitary matrices and the matrix $\hat{\Sigma}_1$ is a $[m\bar{n} + (m-1)(n+1)] \times [m\bar{n} + (m-1)(n+1)]$ diagonal matrix containing the largest singular values of $\hat{S}$ in decreasing order. In equation (5.23) the matrix $\hat{V}_1 \hat{\Sigma}_1^{-2} \hat{V}_1^*$ is the Moore-Penrose pseudoinverse of the best approximation (in Frobenius norm sense of rank $m\bar{n} + (m-1)(n+1)$) of $\hat{S}^* \hat{S}$. Note from (5.14) and (5.19) that the rank of S is $m\bar{n} + (m-1)(n+1)$. Hence, the aforementioned matrix $\hat{V}_1 \hat{\Sigma}_1^{-2} \hat{V}_1^*$ is a consistent estimate of $(S^* S)^{\dagger}$. It follows from the previous discussion that an approximate closed-form minimizer of the criterion (5.7) can be obtained using the following non-iterative algorithm:

(i) Determine an initial consistent estimate of $\{d_{k,j}\}$ by minimizing the criterion in (5.22):

$$\{\hat{d_{k,j}}\} = \underset{\{d_{k,j}\}}{arg\,min}\, \|S^* \hat{\mu}\|^2. \tag{5.25}$$

(ii) Use the estimates of $\{d_{k,j}\}$ obtained in the previous step to compute a consistent estimate $\hat{S}$ of the matrix S. Obtain new estimates of $\{d_{k,j}\}$ by minimizing (5.23).

(iii) (Optional). If the accuracy of the initial estimates obtained in step (i) is suspected to be poor, then re-iterate step (ii) a few times using the most recent estimate of $\{d_{k,j}\}$ to form $\hat{S}$.

Final step Use the latest estimates of $\{d_{k,j}\}$ to estimate $\{c_k\}$ via (5.11).

5.4 COFE-MUSIC

Our discussion of the basic MUSIC approach in this section will necessarily be brief and rather abstract. The reader less familiar with MUSIC should consult, for example, [7][2][10].

We assume that the data consists of P signals superimposed in observation noise. Let M denote the dimension of the "data snapshots". The snapshot vectors are either constructed by the user from a single data string (the time series case) or obtained from an array of M sensors (the array processing case). The MUSIC approach to signal parameter estimation relies on certain orthogonality conditions which in a general form can be written as

$$E^* A(z) = 0, \qquad \text{at and only at} \qquad z = \{\lambda_k\}_{k=1}^{P}. \tag{5.26}$$

In the above equation,

- $\{\lambda_k\}_{k=1}^{P}$ are the signal parameters to be estimated.
- E is an $M \times (M - P)$ unknown matrix whose range space can be consistently estimated from the data snapshots. Note that only the range space of E is of importance for (5.26). An orthonormal basis of the estimated range space is denoted by $\hat{E}$.

- $A(z)$ is either a matrix or more frequently a vector, which has a known application-dependent functional form.

To proceed we need some terminology. We call $\{\lambda_k\}$ the signal poles. We say that we have a linear array (LA) configuration when a typical column of $A(z)$ has the form

$$a(z) = \begin{bmatrix} 1 & z^{k_1} & z^{k_2} & \cdots & z^{k_{M-1}} \end{bmatrix}^T \tag{5.27}$$

where k_j are integers. Uniform Linear Array (ULA) configurations correspond to $k_j = j$ $(j = 1, 2, \cdots, M-1)$; otherwise we say that we have a Non-Uniform Linear Array (NULA) configuration. Note that in this framework, NULA is ULA with "gaps", and that if the signal poles are distinct (and the configuration dealt with is LA) then (see [11][7][2][10]):

$$A(z) = a(z). \tag{5.28}$$

However, if the multiplicity of the signal poles is greater than one, then $A(z)$ has several columns (the number of columns depends upon the multiplicity of the poles). For instance, if all the poles are double then

$$A(z) = \begin{bmatrix} a(z) & d(z) \end{bmatrix} \tag{5.29}$$

where $d(z) = \frac{d(a(z))}{dz}$ ([13]). As an example of such a case, we note that the estimation of parameters in symmetric non-causal autoregressive models by subspace methods leads to a "double pole" problem [13]. It follows from the previous discussion that for LA configurations each element of the matrix product $E^* A(z)$ on the left hand side of (5.26) is a polynomial of degree k_{M-1}. Let $\{B_1(z), B_2(z), \cdots, B_m(z)\}$ denote these polynomials. (Here m is equal to $M - P$ times the number of columns of $A(z)$). As $\{\lambda_k\}_{k=1}^P$ are the unique solutions of the equations $B_k(z) = 0$ $(k = 1, 2, \cdots, m)$, it follows that

$$C(z) = \prod_{k=1}^{P} (z - \lambda_k) \tag{5.30}$$

is the GCD of the set $\{B_k(z)\}$. Hence the estimation of $\{\lambda_k\}$ can be reduced to the COFE problem associated with the set of polynomials $\{\hat{B}_k(z)\}$ derived from $\hat{E}$ in the same way as $\{B_k(z)\}$ are obtained from E. Note that in the present case $\bar{n} = k_{M-1} - P$ as $n = P$.

In summary, the COFE MUSIC algorithm consists of the following steps:

(i) Compute the matrix $\hat{E}$ (in the MUSIC approach this is usually done by the eigendecomposition of the $M \times M$ sample covariance matrix of the data snapshots [7][10]). Construct the vector $\hat{\mu}$ from the rows of $\hat{E}^*$ padded with zeros as necessary.

(ii) Apply the COFE method to $\hat{\mu}$ to obtain estimates of the coefficients of $C(z)$ in (5.30). Derive estimates of $\{\lambda_k\}_{k=1}^{P}$ from the zeros of the estimated polynomial $C(z)$.

5.5 System Identification

Consider an ARARX (auto-regressive auto-regressive exogenous) multi-input single output system described by the equation

$$a(z^{-1})y(t) = \sum_{k=1}^{P} b_k(z^{-1})u_k(t) + \frac{1}{c(z^{-1})}e(t), \tag{5.31}$$

where $\{y(t)\}$ is the output signal, $\{u_k(t)\}$ are input signals, $\{e(t)\}$ is Gaussian distributed white noise, t denotes the time variable, $a(z^{-1})$, $\{b_k(z^{-1})\}$ and $c(z^{-1})$ are the polynomials representing the system, P is the number of input signals present and z^{-1} is the unit delay operator (i.e. $z^{-k}y(t) = y(t-k)$). It is assumed that $a(z^{-1})$ and $\{b_k(z^{-1})\}$ $\forall k$ are coprime polynomials. The system equation (5.31) can be rewritten as

$$\alpha(z^{-1})y(t) = \sum_{k=1}^{P} \beta_k(z^{-1})u_k(t) + e(t), \tag{5.32}$$

with

$$\alpha(z^{-1}) = a(z^{-1})c(z^{-1}) = \sum_{s=0}^{l} \alpha_s z^{-s}, \tag{5.33}$$

and

$$\beta_k(z^{-1}) = b_k(z^{-1})c(z^{-1}) = \sum_{s=0}^{l} \beta_s^{(k)} z^{-s}, \qquad (k = 1, 2, \cdots, P). \tag{5.34}$$

We assume here that we know the degree l of these polynomials, i.e., the system order. The assumption that each polynomial has same order l is made only for notational convenience (as $\beta_s^{(l)}$ can be zero for any s). The problem of system identification consists of estimating the coefficients of $a(z^{-1}), \{b_k(z^{-1})\}$ and $c(z^{-1})$ from the available N data samples of the input and output signals $\{u_k(t)\}_{k=1}^P$ and $\{y(t)\}$.

As is well known (see, e.g. [8]), the ML estimates of the parameters in (5.31) can be obtained by minimizing

$$\sum_{t=1}^{N}[c(z^{-1})(a(z^{-1})y(t) - \sum_{k=1}^{P} b_k(z^{-1})u_k(t))]^2. \tag{5.35}$$

Furthermore, as shown in [12] and [8], the minimization of (5.35) is exactly equivalent to minimizing

$$(\hat{\gamma} - \gamma)^T \hat{P}^{-1}(\hat{\gamma} - \gamma) \tag{5.36}$$

where

$$\gamma = [\alpha_1, \alpha_2, \cdots, \alpha_l, \beta_1^{(1)}, \beta_2^{(1)}, \cdots, \beta_l^{(1)}, \cdots, \beta_1^{(P)}, \beta_2^{(P)}, \cdots, \beta_l^{(P)}]^T, \tag{5.37}$$

$$\hat{\gamma} = \hat{P}\sum_{t=1}^{N}\phi(t)y(t), \tag{5.38}$$

$$\hat{P} = \left(\sum_{t=1}^{N}\phi(t)\phi^T(t)\right)^{-1} \tag{5.39}$$

and where

$$\begin{aligned}\phi(t) = [-y(t-1), \cdots, -y(t-l), u_1(t), \cdots, u_1(t-l),\\ \cdots, u_P(t), \cdots, u_P(t-l)]^T.\end{aligned} \tag{5.40}$$

By noting that (5.36) can be rewritten as

$$(\hat{P}^{-1/2}\hat{\gamma} - \hat{P}^{-1/2}\gamma)^T(\hat{P}^{-1/2}\hat{\gamma} - \hat{P}^{-1/2}\gamma) = (\hat{\nu} - \nu)^T(\hat{\nu} - \nu), \tag{5.41}$$

where $\hat{P}^{1/2}$ is the Hermitian square root of $\hat{P}$, $\hat{\nu} \triangleq \hat{P}^{-1/2}\hat{\gamma}$ and $\nu \triangleq \hat{P}^{-1/2}\gamma$, it can be shown that the present system identification problem is similar to the COFE problem stated in section 3 (see (5.7)). As the polynomials

$\alpha(z^{-1})$ and $\{\beta_k(z^{-1})\}$, whose coefficients make the vector γ have $c(z^{-1})$ as their GCD, γ, can be written as,

$$\gamma = Gc, \tag{5.42}$$

where the matrix G is made from the coefficients of the polynomials $a(z^{-1})$ and $\{b_k(z^{-1})\}$, and the vector c is made from the coefficients of $c(z^{-1})$, similarly to (5.6) and (5.9). From (5.41) we have

$$\nu = \hat{P}^{-1/2}\gamma = \hat{P}^{-1/2}Gc, \tag{5.43}$$

and calculations similar to those in section 3 result in the concentrated criterion function

$$\hat{\nu}^*\Pi^{\perp}_{\hat{P}^{-1/2}G}\hat{\nu}. \tag{5.44}$$

Since S spans the null space of G^* and $\hat{P}^{-1/2}$ is full rank, $\hat{P}^{1/2}S$ spans the null space of $(\hat{P}^{-1/2}G)^*$. Thus, similarly to (5.20) we have the criterion

$$\hat{\nu}^*\Pi^{\perp}_{\hat{P}^{-1/2}G}\hat{\nu} = \hat{\nu}^*\Pi_{\hat{P}^{1/2}S}\hat{\nu} = \hat{\nu}^*\hat{P}^{1/2}S(S^*PS)^{\dagger}S^*\hat{P}^{1/2}\hat{\nu} = \hat{\gamma}^*S(S^*PS)^{\dagger}S^*\hat{\gamma}, \tag{5.45}$$

and similarly to (5.23) the estimates of the coefficients for $a(z^{-1})$ and $b_k(z^{-1})$ can be obtained by

$$\{\hat{a}(z^{-1}), \{\hat{b}_k(z^{-1})\}\} = \underset{\{a(z^{-1}),\{b_k(z^{-1})\}\}}{arg\,min} \; \hat{\gamma}^*S\;(\hat{\hat{V}}_1\hat{\hat{\Sigma}}_1^{-2}\hat{\hat{V}}_1^*)\;S^*\hat{\gamma}, \tag{5.46}$$

where $\hat{\hat{V}}_1$ and $\hat{\hat{\Sigma}}_1$ are obtained by the SVD of $(P^{1/2}S)$ as defined in (5.24), and the matrix $\hat{\hat{V}}_1\hat{\hat{\Sigma}}_1^{-2}\hat{\hat{V}}_1^*$ is the Moore-Penrose pseudoinverse of the best approximation (in Frobenius norm sense of rank $m\bar{n}+(m-1)(n+1)$) of $\hat{S}^*\hat{P}\hat{S}$. Therefore the COFE algorithm can be used for solving the present system identification problem with the modification that (5.46) is used instead of (5.23) in step (ii) of the algorithm.

5.6 Simulation Studies

In Sections 4 and 5 we discussed two applications of the COFE algorithm in signal processing: the problem of signal pole estimation using MUSIC and the problem of parameter estimation in System Identification. In this section we study both applications using numerical examples.

5.6.1 COFE MUSIC

We compare the performance of COFE MUSIC with ROOT MUSIC [2] (the preferred version of MUSIC for LA) and ESPRIT [6]. This comparison is done for the problem of estimating the spatial parameters in array processing, i.e., the estimation of the directions of arrival (DOA) of the signals. We consider ULA and NULA configurations with five sensors (M=5). The spacing between the sensors in the ULA is equal to half the wavelength and the NULA is obtained from an ULA of 6 sensors after removing the 4th sensor (hence $\{k_j = 1, 2, 3, 5, 6\}$). Two uncorrelated white Gaussian sources ($P = 2$) of equal strength are present at 0 and 5 degrees. The noise is assumed to be white Gaussian and the signal-to-noise ratio (SNR) is defined as

$$SNR = 10 \log_{10} \frac{\sigma_s^2}{\sigma_n^2}, \tag{5.47}$$

where σ_s^2 and σ_n^2 are the signal and noise powers respectively. The array covariance matrix is calculated using 100 independent snapshots ($N = 100$). The mean square error (MSE) is used as the measure of the performance, defined as

$$MSE(\hat{\theta}) = \frac{1}{K} \sum_{k=1}^{K} (\theta_{true} - \hat{\theta}^{(k)})^2, \tag{5.48}$$

where θ_{true} is the true DOA and $\hat{\theta}^{(k)}$ is the estimated DOA in the kth realization. Here the MSE is obtained by averaging the outcomes of 100 independent realizations ($K = 100$).

First we compare COFE MUSIC with ROOT MUSIC. The NULA configuration is used to observe the two sources. In this case ESPRIT is not applicable because the array can not be divided into rotational invariant sub-arrays. Figure 5.1 shows the variation of the MSE as a function of SNR for the source at 0 degrees along with the Cramer Rao Bound (CRB). The MSE obtained from two variants of COFE (COFE with and without step (ii)) are shown. (The MSE for the other source behave in a similar manner). It is clear from the figure that at high SNR COFE MUSIC performs similar to ROOT MUSIC. However, COFE MUSIC has a lower threshold value than ROOT MUSIC. In this case we obtain a threshold gain of (almost) 5 dB. The error patterns for both steps of COFE are similar. A gain of (almost) 2 dB is achieved using step (ii), but one should note that step (ii) is computationally

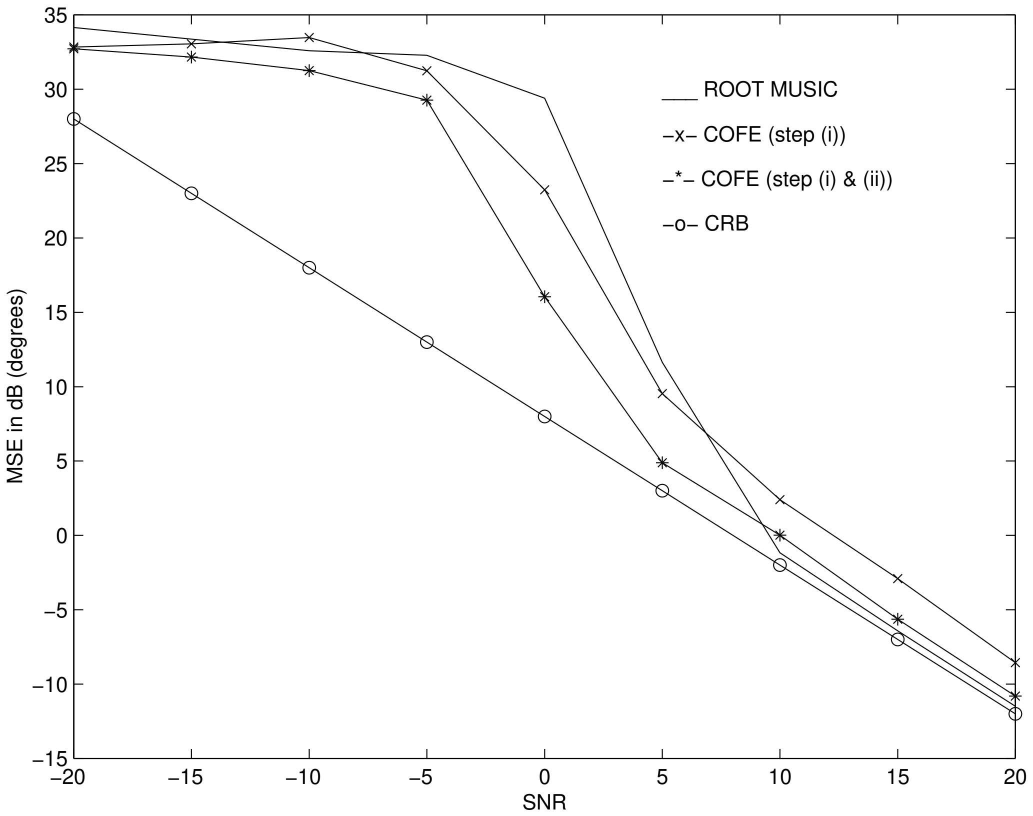

Figure 5.1: MSE of the DOA estimate for the source at 0 degrees and the CRB as functions of the SNR.

expensive. The main advantage of COFE MUSIC over ROOT MUSIC is the ease by which we obtain the estimates. As discussed earlier, unlike ROOT MUSIC, which must obtain the P signal roots from the roots of a M degree polynomial, COFE MUSIC directly obtains the polynomial of degree P (P is the number of sources present) whose roots give the estimates of the desired DOA's.

Next we compare the performance of COFE MUSIC with that of ESPRIT using an ULA. In this case the sources are assumed to have plane wavefronts with decaying or exploding amplitude, i.e., the signal poles do not have unit modulus but are given by $\{\lambda_k = \gamma_k e^{i\pi \sin(\theta_k)}\}$, where γ_k is the amplitude and θ_k is the DOA. For our two sources we use $\gamma_1 = 0.9$ and $\gamma_2 = 1.1$. Note that in many other applications the collected data are modeled similarly, e.g., for damped sinusoids, distributed sources [1] and magnetic resonance spectroscopy [14]. ROOT MUSIC is not applicable (as the signals poles do not have unit modulus), but the signal parameters can be obtained using ESPRIT or COFE MUSIC. Figure 5.2 depicts the MSE for the estimates of the DOA for the source at 0 degrees as a function of the SNR. It is clear that

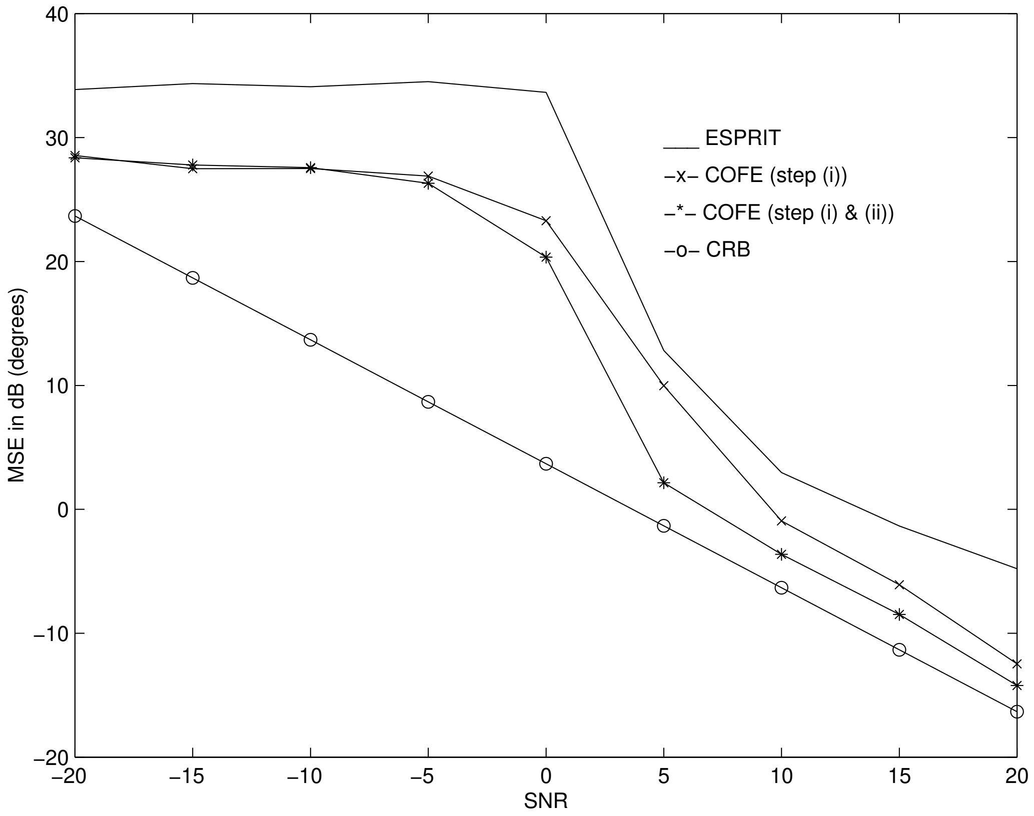

Figure 5.2: MSE of the DOA estimate for the source at 0 degrees and the CRB as functions of the SNR.

COFE MUSIC performs better than ESPRIT for all SNR values considered. Moreover, the MSE for the estimates obtained by COFE MUSIC approaches the CRB (as the SNR increases) as was also the case in the previous example.

The real need of COFE MUSIC is seen when we have a NULA observing signals of decaying or exploding amplitude as in the previous example. Here, neither ROOT MUSIC nor ESPRIT are applicable. For ROOT MUSIC it becomes impossible to separate the signal poles from the noise poles, and for ESPRIT the requirement for sub-array invariance is not fulfilled. Figure 5.3 shows the behavior of the MSE as a function of the SNR for the DOA estimates of the source at 0 degrees. Comparing Figure 5.1 with Figure 5.3 it

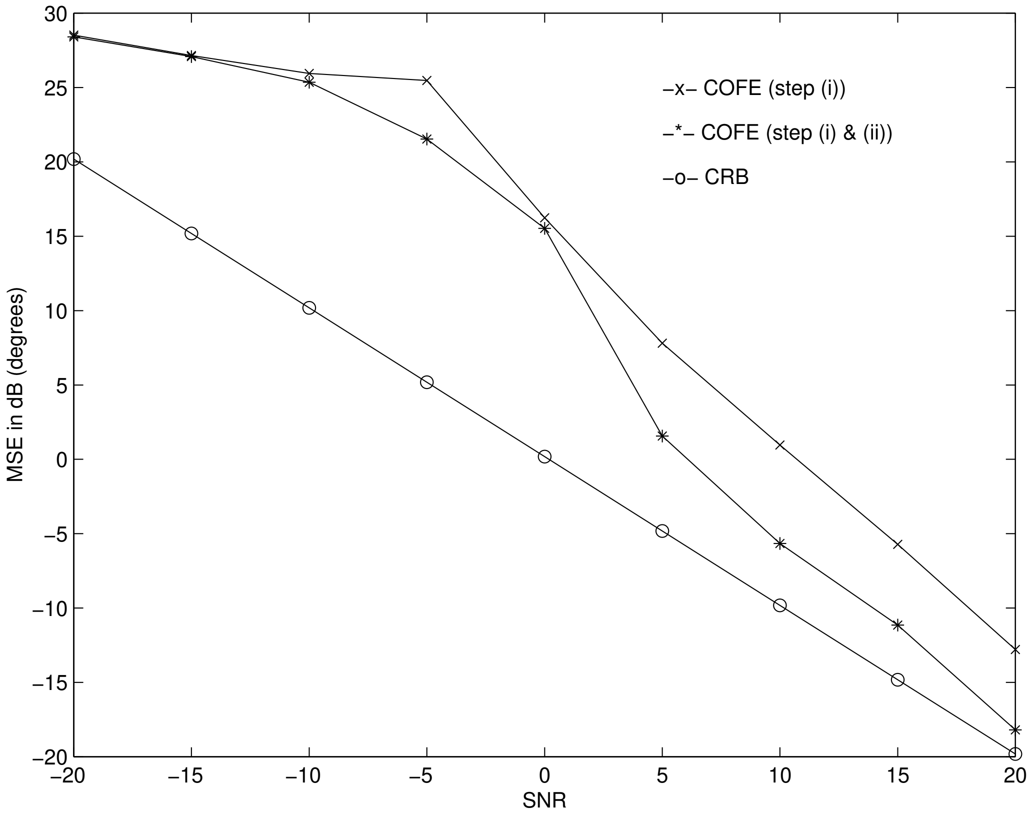

Figure 5.3: MSE of the DOA estimate for the source at 0 degrees and the CRB as functions of the SNR.

can be seen that COFE MUSIC performs similarly in both situations and also that it achieves CRB asymptotically in both cases.

5.6.2 System Identification

Next we consider an ARARX system with two inputs ($P = 2$), described by

$$a(z^{-1}) = 1 - 1.5z^{-1} + 0.7z^{-2} \tag{5.49}$$

$$b_1(z^{-1}) = z^{-1} + 0.5z^{-2} \tag{5.50}$$

$$b_2(z^{-1}) = z^{-1} - 0.5z^{-2} \tag{5.51}$$

$$c(z^{-1}) = 1 - 1.0z^{-1} + 0.2z^{-2}. \tag{5.52}$$

The input sequences $\{u_k(t)\}$ are white Gaussian signals, independent of each other and also of the white Gaussian noise sequence $\{e(t)\}$. The SNR of the system is defined as

$$SNR = 10 \log_{10} \frac{\sum_t [\sum_{k=1}^{P} \frac{b_k(z^{-1})}{a(z^{-1})} u_k(t)]^2}{\sum_t [\frac{1}{a(z^{-1})c(z^{-1})} e(t)]^2}. \tag{5.53}$$

In the simulations we have used 500 data samples ($N = 500$) in each simulation and the results are averaged over 100 independent realizations. The COFE algorithm is compared to the standard predictor error method (PEM). The PEM estimates are obtained using the standard MATLAB command "PEM" (from the system identification toolbox [5]). Figure 5.4 shows the sum of the MSEs for the estimated parameters of the system as a function of the SNR for both COFE and PEM. The performance of COFE is marginally better than that of PEM at high SNR but at low SNR PEM performs better than COFE. Once again the second step of the COFE algorithm improves the estimates by almost 2 dB.

A good thing to note about COFE is that it is simpler computationally than PEM. For the estimation problem studied here COFE on the average requires 480k flops in MATLAB with one step and 690k flops with two steps, whereas PEM required 2107k flops.

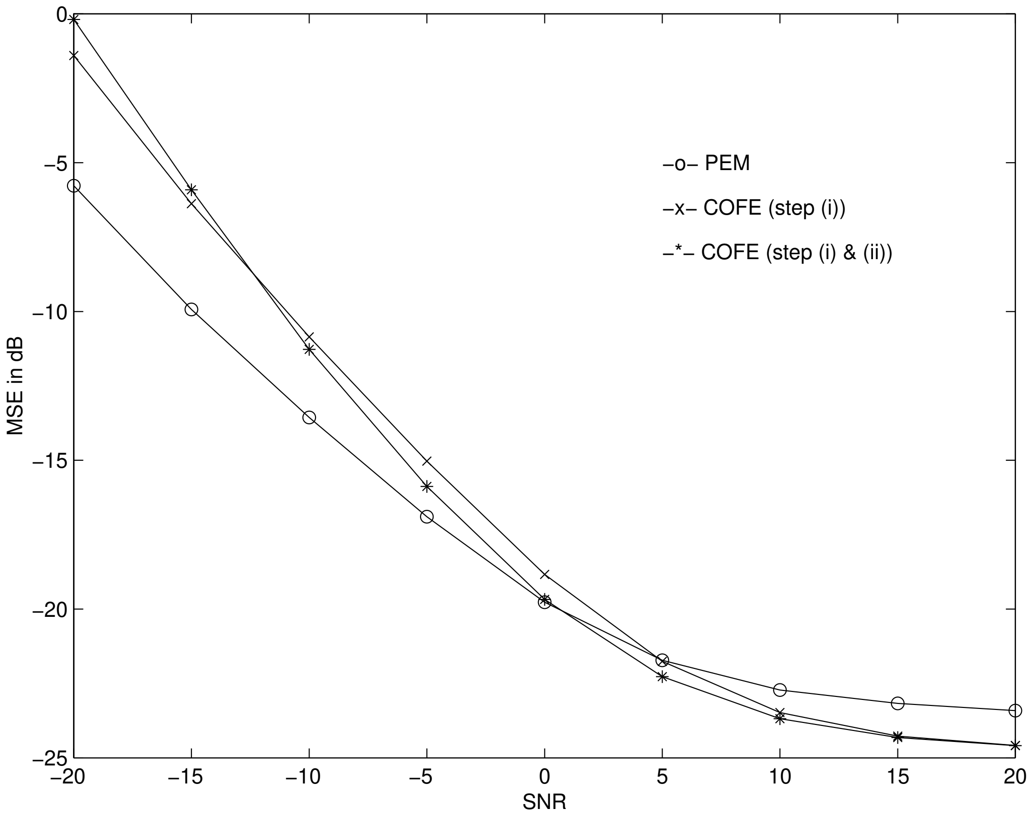

Figure 5.4: Sum of MSE in the estimation of the system parameters as a function of the SNR.

Bibliography

[1] Asztely D, Ottersten B. Modified Array Manifold for Signal Waveform Estimation in Wireless Communications. In *Signals, Systems and Computers, 1996. Conference Record of the 30th Asilomar*, volume 1, pages 738–741, 1997.

[2] Barabell AJ. Improving the Resolution Performance of Eigenstructure Based Direction Finding Algorithm. In *Proceedings ICASSP'83*, pages 336–339, Boston, MA, 1983.

[3] Hua Y. Fast Maximum Likelihood for Blind Identification of Multiple FIR Channels. *IEEE Transactions on Signal Processing*, 44:661–672, March 1996.

[4] Li Y, Ding Z. Blind Channel Identification Based on Second-Order Cyclostationary Statistics. In *Proceedings of ICASSP'96*, pages 81–84, Minneapolis, April 1996.

[5] Ljung L. *System identification toolbox user's guide, version 5.* The MathWorks Inc., 3 Apple Hill Drive, Natick, MA, USA, 2001.

[6] Roy R, Kailath T. ESPRIT - Estimation of Signal Parameters via Rotational Invariance Techniques. *IEEE Transactions on Acoustic Speech and Signal Processing*, 37:984–995, July 1989.

[7] Schmidt RO. *A Signal Subspace Approach to Multiple Emitter Location and Spectral Estimation.* PhD thesis, Stanford University, Stanford, CA, 1989.

[8] Söderström T, Stoica P. *System Identification.* Prentice-Hall, International, UK, 1989.

[9] Söderström T, Stoica P, Friedlander B. An Indirect Prediction Error Method. *Automatica*, 27:183–188, Jan 1991.

[10] Stoica P, Nehorai A. MUSIC, Maximum Likelihood and Cramer-Rao Bound. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37:720–741, May 1989.

[11] Stoica P, Sharman KC. Maximum Likelihood Methods for Direction-of-Arrival Estimation. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38:1132–1143, July 1990.

[12] Stoica P, Söderström T. Common Factor Detection and Estimation. *Automatica*, 33:985–989, 1997.

[13] Stoica P, Sorelius J. Subspace-Based Parameter Estimation of Symmetric Non-causal Autoregressive Signals from Noisy Measurements. *IEEE Transactions on Signal Processing*, 47:321–331, 1999.

[14] Stoica P, Sundin T. Exact ML Estimation of Spectroscopic Parameters. *Journal of Magnetic Resonance*, 145:108–114, 2000.

[15] Xu G, Liu H, Tong L,Kailath T. A Least-Squares Approach to Blind Channel Identification. *IEEE Transactions on Signal Processing*, 43:2982–2993, Dec 1995.

Chapter 6

Data-dependent broad-band beamforming with many constraints using few sensors

Data-dependent broad-band beamforming with many constraints using few sensors

Abstract

A common technique for performing data-dependent beamforming is to minimize the beamformer output power in a constrained manner. In real beamforming applications the number of available degrees of freedom is often limited due to the limited number of sensors. If the number of constraints is numerous the emphasis of the beamformer will be to fulfill the constraints and the interferers may therefore not be sufficiently suppressed. In this paper we propose a new approach where we include the constraints in the cost function for the optimization of the beamformer. It turns out that the beamformer obtained by this approach performs better than the standard beamformer in the case of few array sensors while performing equally well for more sensors in the array.

6.1 Introduction and problem formulation

A common technique to enhance a signal of interest using sensor arrays is beamforming [4][1][5], in which the samples collected by the sensors are combined in a manner such that the signal of interest is enhanced and other interfering signals are suppressed. Usually this is accomplished by forming a beam in the look-direction (towards the signal source of interest) where everything within the angular width of the beam is included and the rest is suppressed. The beamformer can be narrowband or broadband, depending on the nature of the signal of interest. In this paper we will focus on the broadband case. A common problem with many beamforming techniques is that the beam is very narrow. If it is perfectly focused on the source of interest (i.e. we know the exact look direction) there is no problem since then we obtain the signal from the source of interest with minimum distortion. Usually, however, the direction of arrival (DOA) of the signal of interest is known only to within some angular tolerance and if the beam is very narrow we may suppress the signal of interest instead of enhancing it, a phenomenon called target signal cancellation. One approach to overcome this problem is to broaden the beam of the beamformer. In [2] derivative constraints were successfully imposed to broaden the beam. By constraining the power response of the beamformer in the look direction to have derivatives (with respect to angle and up to a given order) equal to zero, a maximally flat beamformer response over a finite region of the beam was achieved.

In data-dependent beamforming interfering signals and noise sources are suppressed by using the available degrees of freedom to place nulls in the corresponding directions. If, however, the available degrees of freedom in the beamformer are not sufficient to do this the performance of the beamformer will become worse since the interfering signals are not sufficiently suppressed. Additional constraints on the beamformer, such as the derivative constraints in [2], reduce the number of available degrees of freedom even further. If the number of sensors in the array is limited, which is often the case in real scenarios, the inclusion of additional constraints may for this reason make the beamformer perform worse than it would without the use of the additional constraints.

The approach in this paper is to loosen the requirements on the beamformer to fulfill the constraints imposed on it. Instead of exactly fulfilling all the constraints we include the constraints into the cost function. Thus we obtain the beamformer by solving an unconstrained minimization problem. In standard beamforming we may be wasting some of the degrees of freedom for exactly fulfilling the constraints. These extra degrees of freedom could instead be used to suppress the interferers and thereby improve the performance of the beamformer. Numerical examples show indeed that our approach actually results in a performance gain for some scenarios.

We will call the unconstrained beamformer proposed in this paper for the Costfunction-Included-Constraints-beamformer (CIC-beamformer) and the beamformer already available in the literature for the "standard beamformer".

6.2 The CIC-beamformer

In Figure 6.1 the structure of a general broadband array beamformer is shown. L sensors are present in the array and each sensor has a J element tapped delay line attached with time delays T that has the function of coping with the broadband nature of the data. The time delays T_i are broadband

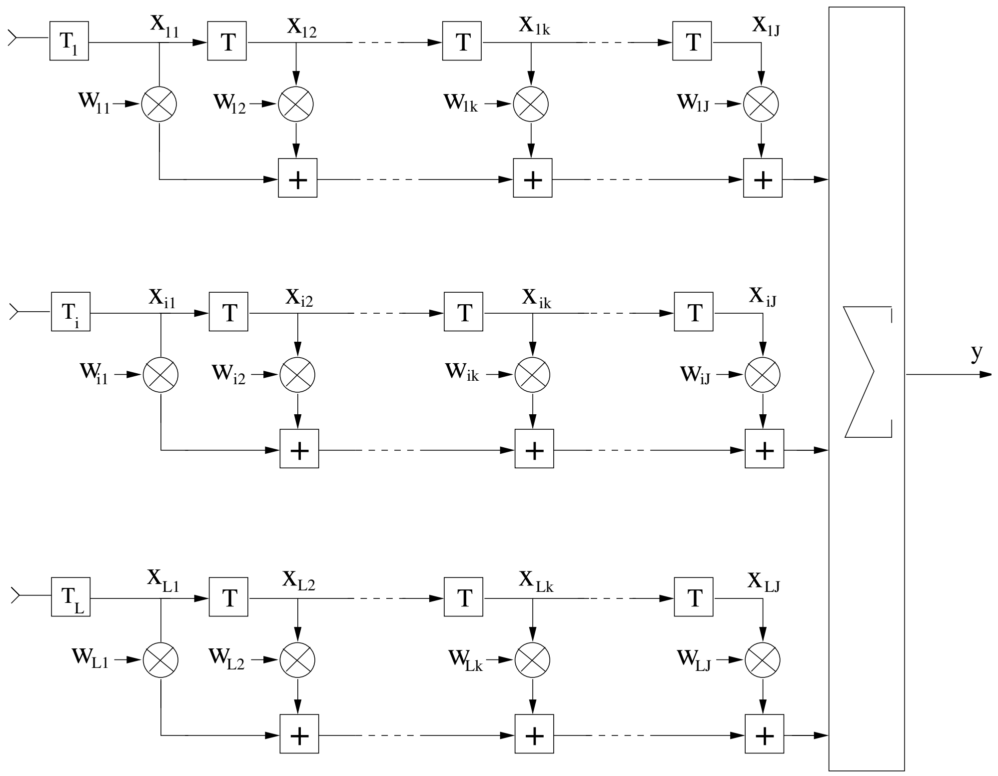

Figure 6.1: A schematic figure showing the arrangement of the data samples and weights in a typical broad-band array beamformer.

delays used to presteer the array in the look direction (which is the DOA of the source of interest), as is explained in [2]. The beamformer weights are denoted by w_{kl} according to Figure 6.1 and the beamformer inputs are denoted by $x_{kl}(t)$. The beamformer output $y(t)$ is then simply

$$y(t) = \sum_{k=1}^{L} \sum_{l=1}^{J} w_{kl} x_{kl}(t) \triangleq w^H x(t), \tag{6.1}$$

where the beamformer weight vector w is defined as

$$w = [w_{11}^H \, w_{21}^H \, \dots \, w_{L1}^H \, w_{12}^H \, \dots \, w_{L2}^H \, \dots \, w_{1J}^H \, \dots w_{LJ}^H]^H \tag{6.2}$$

and the beamformer input vector $x(t)$ is defined in a similar manner. Note that $(\cdot)^H$ stands for complex conjugate transpose.

To obtain the optimum weight vector of the beamformer the standard way ([2]) is to solve a constrained minimization problem in which the expected beamformer output power

$$E[y^*(t)y(t)] = E[w^H x(t)x^H(t)w] = w^H E[x(t)x^H(t)]w \triangleq w^H Rw \qquad (6.3)$$

is minimized subject to different constraints, the most common one being that the beamformer power response in the look direction should be unity. The constrained optimization problem can thus be written as

$$w_{opt}^{standard} = arg\,\min_{w} w^H Rw, \quad \text{s.t.} \quad D^H w = f, \qquad (6.4)$$

where the full rank matrix D and the vector f define the constraints. This minimization problem is easily solved using the method of Lagrange multipliers [3] and the optimized beamformer weight vector is given by

$$w_{opt}^{standard} = R^{-1}D(D^H R^{-1}D)^{-1}f. \qquad (6.5)$$

If D has many columns, meaning that the number of constraints imposed on the optimization is large, the number of degrees of freedom required to fulfill the constraints will also be large, possibly leaving few degrees of freedom for the minimization of the output power of the beamformer. This may result in bad performance of the beamformer, especially in the case when few sensors are present in the array or when the minimization of the beamformer output power is hard, such as when many interferers are present.

The approach in this paper is to fulfill the constraints in (6.4) only approximately, therefore possibly getting a beamformer that performs better in the case when the number of available degrees of freedom is low. More precisely our approach is to merge the constraints into the cost function and minimize the function

$$w_{opt}^{CIC} = arg\,\min_{w} \{w^H Rw + \alpha \|D^H w - f\|^2\}, \qquad (6.6)$$

to obtain the weight vector. In (6.6) α is a scalar constant and R, D, w and f are defined as above. By adjusting α we can control how much weight should be given to the constraints relative to the output power in the minimization of (6.6). The minimum of (6.6) is easily obtained by noting that

$$\begin{aligned} w^H Rw + \alpha \|D^H w - f\|^2 = (w - (R + \alpha DD^H)^{-1}\alpha Df)^H(R + \alpha DD^H) \times \\ \times (w - (R + \alpha DD^H)^{-1}\alpha Df) + \text{const}. \end{aligned} \qquad (6.7)$$

The cost function (6.6) is then minimized by

$$w_{opt}^{CIC} = (R/\alpha + DD^H)^{-1} Df, \tag{6.8}$$

and this is what we call the CIC-beamformer.

Note that the array covariance matrix R is unavailable to us. The best we can do is to estimate R by

$$\hat{R} = \frac{1}{N} \sum_{t=1}^{N} x(t) x^H(t), \tag{6.9}$$

where N is the number of array data snapshots available. Thus, when using the beamformers (6.5) and (6.8) we use the estimated $\hat{R}$ instead of R.

The choice of the user-parameter α in (6.8) is important and below we will give some guidelines for choosing α. By looking at the cost function for the CIC-beamformer (6.6) we can easily see that if there are no interferers, and the Signal to Noise Ratio (SNR) is high, then the first term in (6.6) (that minimizes the noise and the interferers) is of less importance than the second term which is directly proportional to α. This will result in optimal performance for high values of α. For low SNR, however, minimization of the first term in (6.6) is important in order to improve the SNR and optimal performance will be achieved for lower values of α. Furthermore, note that for high SNR it is not critical to choose exactly the optimal α since the emphasize will be on the second term in (6.6) provided that α is chosen reasonably high. These general guidelines regarding the choice of α are confirmed in the numerical examples which show that the choice of α is relatively easy, provided that the SNR and the Signal to Interference Ratio (SIR) are reasonably high.

6.3 Numerical examples

Before going on with the simulations results it is important to note when the CIC-beamformer should be used and when it can be expected to outperform the standard beamformer. The main motivation for the CIC-beamformer is that it compromises between fulfilling the beamformer constraints and suppressing interfering signals. When the number of sensors in the array is high, or when there are no interferers (or the interferers are very weak) this compromise is not needed since then the beamformer will have enough degrees of freedom to both fulfill the constraints and suppress the interfering signals. However, when there are strong interferers and the number of sensors in the array is relatively low, then the CIC-beamformer can be expected to outperform the standard beamformer.

To demonstrate the behavior of the CIC-beamformer we perform Monte-Carlo simulations for several different scenarios and compare the performance of the CIC-beamformer to that of the standard beamformer. As beamformer constraints we use the aforementioned derivative constraints that were presented in [2]. Both first and second order derivative constraints are used. The purpose of the derivative constraints is, as was mentioned earlier, to broaden the beam so that uncertainties about the DOA of the desired signal will not result in target signal cancellation. To make this discussion short we refer to [2] for a more detailed description and derivation of the derivative constraints. We use a broadband uniform linear array (ULA) with the source of interest (for simplicity) located at the broadside (zero degrees) of the array, which means that no presteering is required. The sources are assumed to be far-field so all wavefronts impinging on the array can be considered to be planar. In all the examples the noise is generated as spatially and temporally uncorrelated white Gaussian and the source signals are generated as uncorrelated wideband white Gaussian sequences of powers equal to 1 and with a bandwidth spanning from $0.1f_s$ to f_s, where f_s is the sampling frequency. The order of the tapped delay lines attached to all sensors in the beamformer is $J = 3$ which was found to be sufficient to cope with the wideband characteristics of the signals generated for the problem. For all calculations 100 array snapshots were used in calculating $\hat{R}$. The results presented here are all obtained by averaging the results of 100 independent Monte Carlo simulations. To display the performance of the methods we use the Normalized Mean Squared Error (NMSE)

$$\mathrm{NMSE} = \frac{1}{\sum_{k=1}^{N} \|y(k)\|^2} \sum_{k=1}^{N} \|y(k) - \hat{y}(k)\|^2, \tag{6.10}$$

where $y(k)$ is the signal from the source of interest and $\hat{y}(k)$ is the beamformer output corresponding to the kth array snapshot. The SNR and SIR are defined as

$$\mathrm{SNR} = 10\log_{10}(\frac{P_s}{P_n}), \quad \mathrm{SIR} = 10\log_{10}(\frac{P_s}{P_i}) \tag{6.11}$$

where P_s is the power of the signal from the desired source, P_n is the power of the noise and P_i is the power of the interferers (that always are generated to have the same powers).

To verify that the CIC-beamformer is indeed giving a performance gain compared to the standard beamformer, as a first example we study a scenario with two interfering sources, each with power equal to the source of interest, located at angles of 50 and -50 degrees relative to broadside. At a fixed SNR of 10 dB the number of sensors in the array is varied and the NMSEs of the CIC-beamformer and the standard beamformer are computed. The results are displayed in Figure 6.2 where the NMSEs of the beamformers are plotted as a function of the number of sensors L in the array. It is clear that the CIC-beamformer outperforms the standard beamformer for fewer sensors, whereas the difference between the standard and the CIC-beamformers gets lower with increasing number of sensors. This is to be expected since the number of degrees of freedom available for suppressing the interferers is too low for the standard beamformer to both fulfill the constraints and suppress the interferers when the number of sensors in the array is low. Note that in obtaining the NMSEs, we use a fixed value of $\alpha = 10$ in (6.8). This choice of α was motivated by the figure below for optimal values of α.

To further compare the performance of the CIC-beamformer to that of the standard beamformer we again simulate a scenario with 5 sensors but this time with no interferers and compute the NMSE of the outputs for the two beamformers for different SNR. The results are shown in Figure 6.3 where the NMSEs for the beamformers are plotted as a function of the SNR. As can be seen the CIC-beamformer again outperforms the standard beamformer which is to be expected since the standard beamformer does not have enough degrees of freedom to minimize the noise. Note that in obtaining these results the optimal value of α for each SNR was used (since the SNR was varied we could not choose a fixed value for α). Also note that the plot in the figure has been truncated.

As we discussed in the previous section the optimal value of α is heavily dependent on the SNR and the SIR. To examine this dependency and verify the rough guidelines we gave in the previous section for choosing α we simulate a scenario with two interfering sources located at 50 and -50 degrees relative to broadside and find the optimal values of α for different SIR. We also simulate the scenario when the interfering sources are located at 40 and -40 degrees, and 30 and -30 degrees. For the beamforming we use a sensor array with 5 sensors (for which the CIC-beamformer in the previous scenario

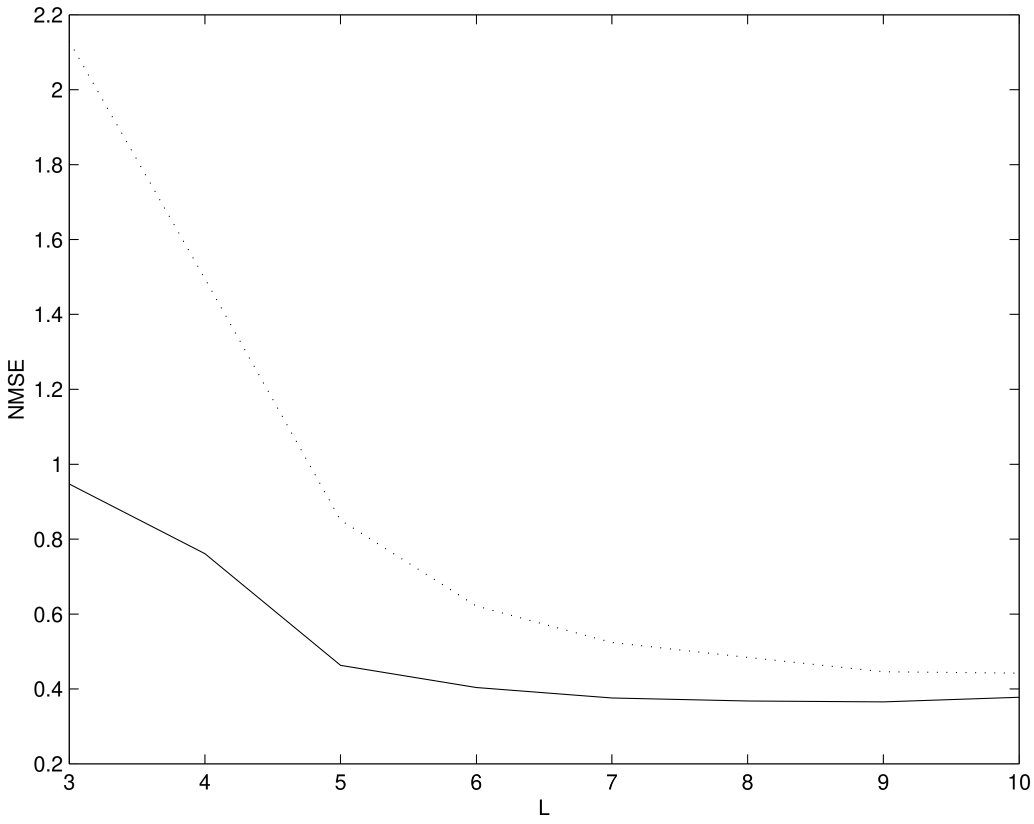

Figure 6.2: NMSE as a function of the number of array sensors in for the standard beamformer (dotted) and the CIC-beamformer (solid). Interfering sources are located at 50 and -50 degrees.

is shown to perform better than the standard beamformer). The SNR is kept at 10 dB. The results are displayed in Figure 6.4, where the optimal values of α for the three sets of interferers (optimal in the sense that it gives the lowest NMSE) are plotted as a function of SIR. As we can see from the figure the optimal values for α is practically constant and equal to 1 for low SIR but as soon as the SIR becomes 0 dB the optimal value for α gets very large.

To study how robust the beamformer is to errors in choosing the optimal value for α we simulate a system with 5 sensors and no interferers and compute the beamformer output for different choices of α. The results are shown in Figure 6.5 where the NMSE of the beamformer output is plotted

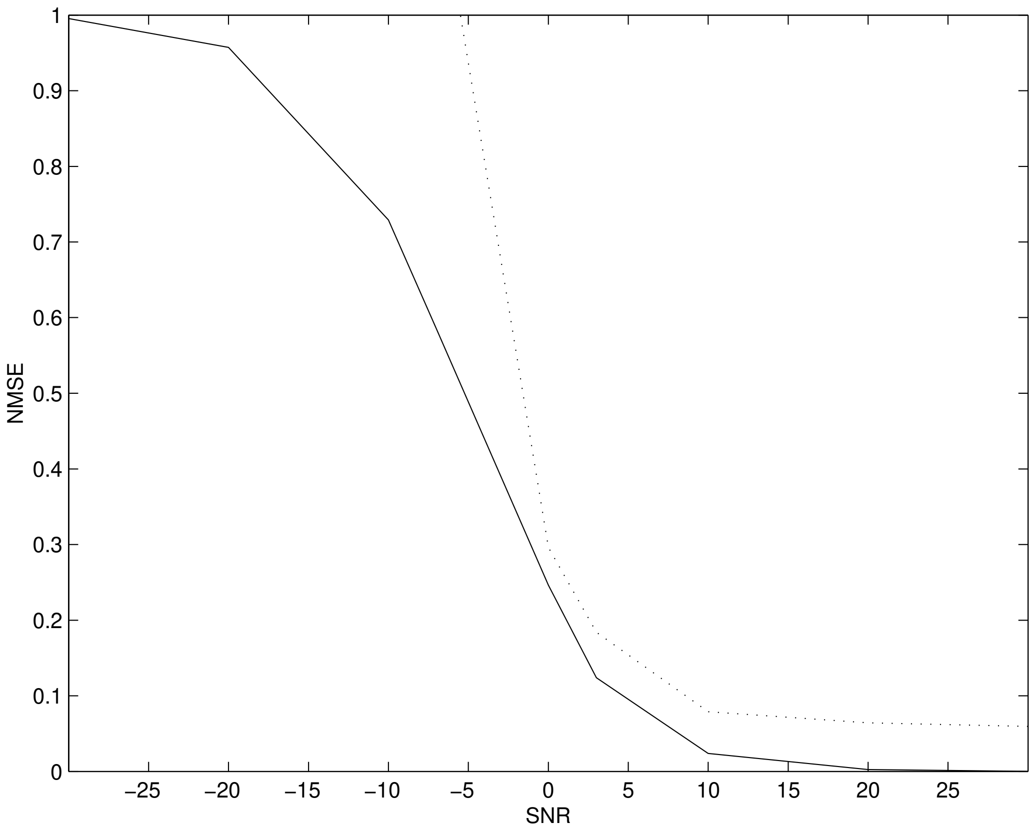

Figure 6.3: NMSE for the output of the standard beamformer (dotted) and the CIC-beamformer (solid), as a function of the SNR for an array with $L = 5$ sensors and no interferers present.

as a function of α for SNRs of -10 dB, 10 dB and 30 dB. In the previous section we stated that it should be more critical for low SNR (and SIR) than for high SNR that the chosen value of α is close to the optimal choice. This is confirmed in the figure where the curve for the NMSE has a very wide minimum for high SNR (10 dB and 30 dB), while it has a narrow minimum for low SNR (-10 dB).

Finally we want to verify that the broadband constraints are being sufficiently fulfilled for the CIC-beamformer. To this end we study a scenario where two interferers of power equal to the source of interest are present at -50 and 50 degrees and the DOA of the source of interest is varied. We keep,

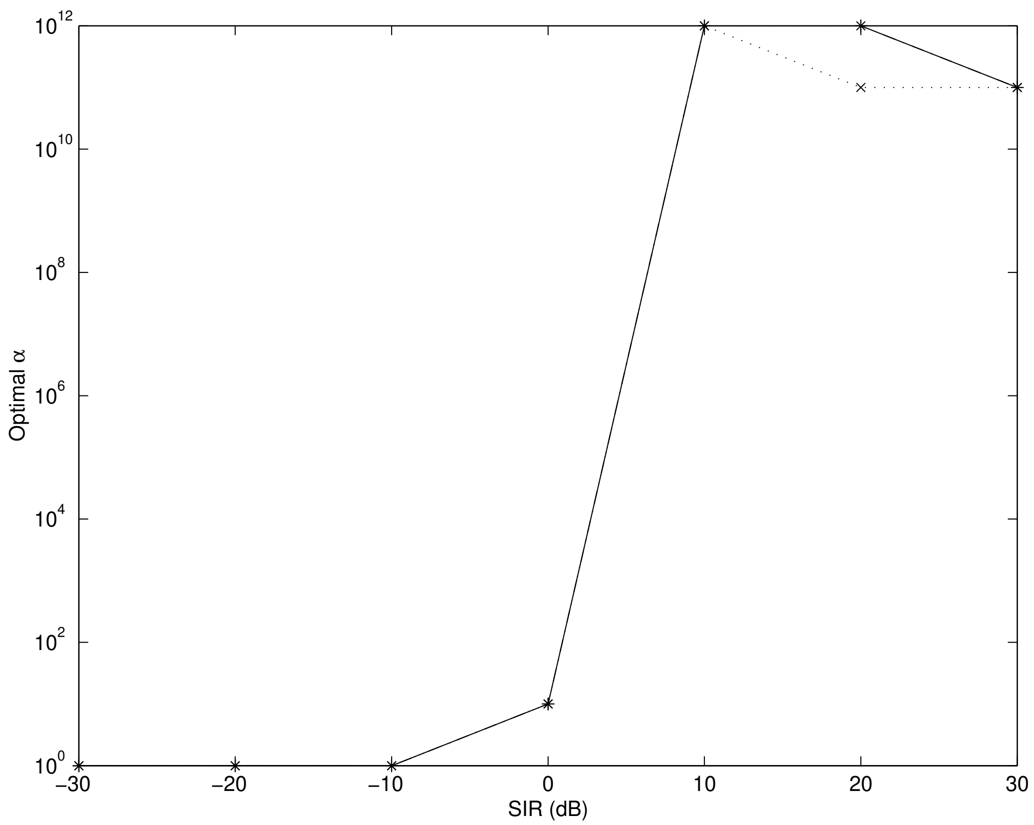

Figure 6.4: Optimal alpha values as a function of SIR with two interferers present at 50 and -50 degrees (+), 40 and -40 degrees (*), and 30 and -30 degrees (x).

however, the beamformer look direction at broadside to simulate an uncertainty in the DOA of the source of interest. For the standard beamformer this will cause signal cancellation of the signal of interest. A $L = 5$ sensor array is used to observe the sources for an SNR of 10 dB. Three different beamformers are applied to the data: the standard beamformer with derivative constraints [2], the CIC-beamformer with derivative constraints and the standard beamformer without any derivative constraints. The results are presented in Figure 6.6 where the NMSE of the beamformer output is plotted as a function of the direction (θ) to the source of interest. It is clearly seen that the standard beamformer without derivative constraints performs

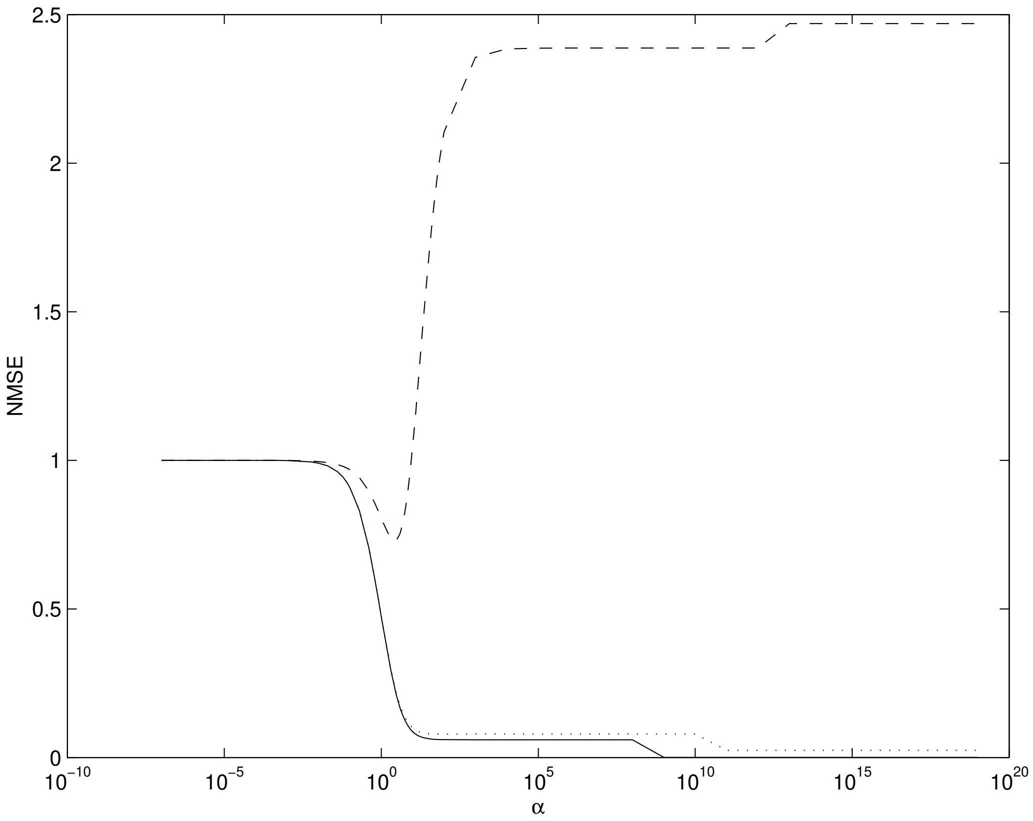

Figure 6.5: NMSE for the output of the CIC-beamformer as a function of α for a SNR of -10 dB (dashed), 10 dB (solid) and 30 dB (dotted), for an array with $L = 5$ sensors. No interferers are present.

best when the beam is focused on the source of interest, as is to be expected. It is also clear, however, that when the beam is not focused on the source of interest the CIC-beamformer performs better, or similarly, to the standard beamformer without derivative constraints. The standard beamformer with derivative constraints clearly performs worse than the other two beamformers. Thus we can conclude that the derivative constraints are working and that the CIC-beamformer is the best performing beamformer for this scenario.

6.4 Concluding remarks

In this paper we have presented a new constrained data-dependent beamformer that works well for the case when the number of sensors in the beamforming array is low and the number of constraints on the beamformer is high, which is not true for standard data-dependent beamformers. This is especially true for low SNR and SIR, and the superior performance of the presented beamformer is validated in numerical examples. Guidelines are given for the choice of the user parameter in the presented beamformer and these are confirmed by the numerical examples. Thus the presented beamformer should be the beamformer to choose when the number of sensors in the array is low and the SIR and SNR are low.

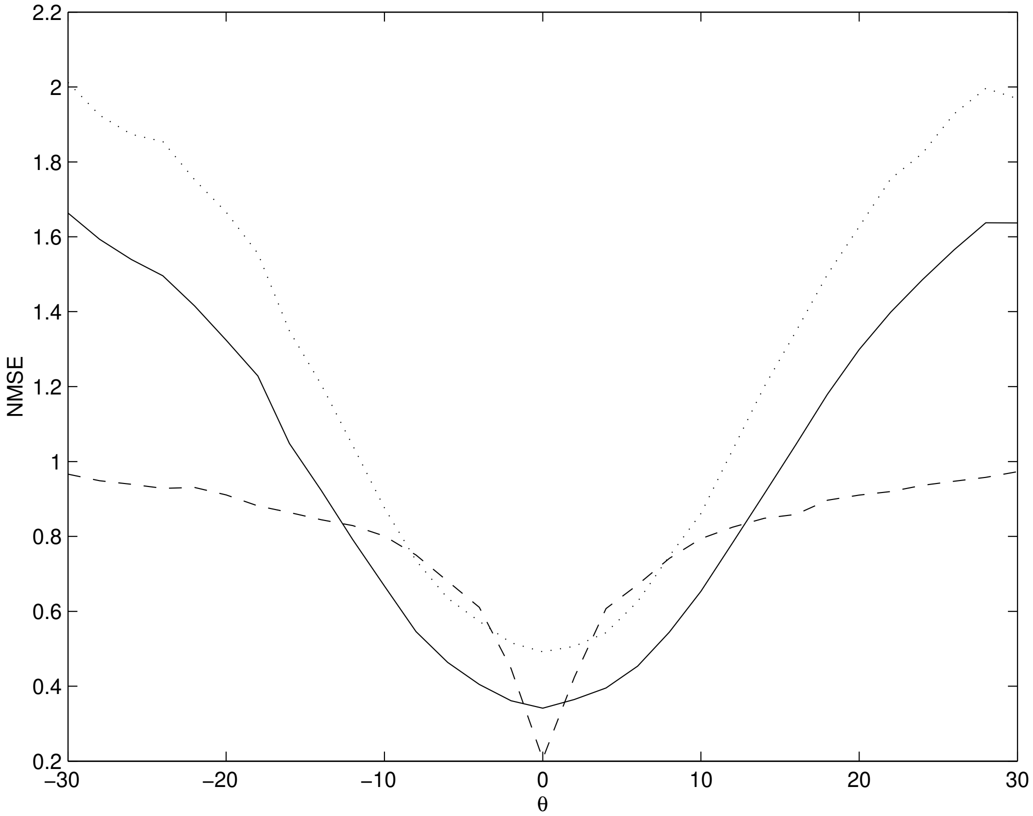

Figure 6.6: NMSE for the output of the standard beamformer without any derivative constraints (dashed), the standard beamformer with derivative constraints (dotted) and the CIC-beamformer (with derivative constraints) (solid), as a function of the direction θ to the desired source for an array with $L = 5$ sensors and interferers present at 50 and -50 degrees.

Bibliography

[1] Capon J. High Resolution frequency-wavenumber spectrum analysis. *Proceedings of IEEE*, 57:1408–1418, Aug 1969.

[2] Er MH, Cantoni A. Derivative constraints for broad-band element space antenna array processors. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 31(6):1378–1393, Dec. 1983.

[3] Simmons GF. *Differential equations with applications and historical notes*. McGraw-Hill, Inc., 2rd edition, 1991.

[4] Vanveen BD, Buckley KM. Beamforming: A versatile approach to spatial filtering. *IEEE ASSP Magazine*, pages 4–24, Apr 1988.

[5] Widrow B, Mantey PE, Griffiths LJ, Goode BB. Adaptive Antenna Systems. *Proceedings of IEEE*, 55:2143–2159, Dec 1967.

Chapter 7

Array Processing for Signals with Non-Zero Means in Colored Noise Fields

Array Processing for Signals with Non-Zero Means in Colored Noise Fields

Abstract

Array processing in spatially correlated noise fields is a difficult problem which cannot be solved without constraining either the noise or the signals in various ways. In this paper we assume that the signals impinging on the array have non-zero means. We show that the problem of array processing in completely general spatially colored noise fields is solvable under the previous assumption. In fact we argue that the non-zero mean signal assumption is probably one of the simplest and least restrictive conditions which makes array processing in colored noise possible. We also present a maximum-likelihood (ML) methodology for signal detection and parameter estimation. The derivation of the ML processor under the aforementioned assumption is conceptually straightforward and its implementation is rather standard. We provide a statistical analysis of the ML signal parameter estimator along with an expression for the associated Cramer-Rao bound (CRB) matrix. Finally, we include some numerical examples to show the type of performance that can be achieved by the estimation and detection approach proposed in this paper.

7.1 Introduction

Array processing in spatially colored (or correlated) noise fields is a practically relevant and theoretically challenging problem that cannot be solved without making certain assumptions on either the noise or the signals impinging on the array, such as:

- The noise is assumed to satisfy a parametric model (for instance a spatial autoregressive equation) [2][5][16].
- The array comprises two sufficiently separated subarrays, the noise vectors of which are uncorrelated with one another [15].

- The temporal correlation length of the signals is (much) larger than that of the noise [10][12].
- The signals are known, possibly up to a multiplicative constant [3].
- The signals are linear combinations of some known basis functions [13].

Any of the previous assumptions may fail to hold in a given application. In fact the set of applications that satisfy at least one of the assumptions above is rather limited.

In this paper we assume that:

- The signals impinging on the array have nonzero means.

We show that under this assumption the array processing problem is solvable without imposing any condition on the spatial noise correlation function. Furthermore, we show that the derivation of the ML processor for detection and signal parameter estimation is straightforward and its implementation is standard. The statistical analysis of the ML parameter estimator, including the derivation of the associated CRB, is also relatively simple.

The non-zero mean signal (NMS) assumption appears to be one of the simplest and probably least restrictive conditions that makes array processing in general colored noise fields possible. In certain applications, such as localizing underground objects by using a gradiometer array [4], nonzero mean signals appear in a natural way. In other applications, such as in communications, the NMS condition can in principle be satisfied. In such applications we can focus on certain signals by simply setting the means of the other (undesired) signals to zero (as explained in some detail later on), which may be a non-negligible advantage in today's crowded communication environment. The NMS condition appears to be feasible in many other applications as well. For more arguments lending support to the NMS assumption we refer to [4]. The latter paper has also treated the problem of signal parameter estimation under the NMS condition. However in [4], unlike in the present paper, the nonzero means of the signals were considered to be a nuisance rather than a quality of the data. Additionally the approach taken in [4] has resulted in a somewhat complex parameter estimation methodology, in spite of the fact that it was limited to uniform and linear arrays and to a special class of noise fields. The approach of this paper is comparatively more general and more straightforward both theoretically and computationally.

Finally, we remark on the fact that the NMS scenario can be shown after some manipulations to be a special case of the scenarios considered in [3] (where the signals were assumed to be known up to some multiplicative constant) and [13] (where the signals were assumed to be linear combinations of some known basis functions). This observation would allow us to obtain some of the results in this paper by specializing the corresponding results in [3] and [13]. However, doing so would make this paper less self-contained and moreover would not necessarily be much simpler than deriving these results directly, which is the approach we take in the following sections.

7.2 Array model and parameter identifiability

Let $y(t) \in \mathcal{C}^{m\times 1}$ denote the tth snapshot of the output of the array. Henceforth m denotes the number of sensors in the array. In the case of narrowband processing and under the planar wave assumption the vector $y(t)$ satisfies the following standard equation:

$$y(t) = Ax(t) + e(t), \tag{7.1}$$

where $x(t) \in \mathcal{C}^{n\times 1}$ is the signal vector (n is the number of signals), $e(t) \in \mathcal{C}^{m\times 1}$ is the noise vector, and

$$A = [a(\theta_1) \dots a(\theta_n)]. \tag{7.2}$$

In (7.2), $a(\theta_k)$ is the direction vector corresponding to the kth signal, and θ_k is the direction-of-arrival (DOA) of that signal. We omit the dependence of A on $\{\theta_k\}$ to simplify the notation. The functional form of the direction vectors is assumed to be known (in other words we assume that the array is calibrated). Additionally the array is assumed to be "unambiguous" in the sense that for any set of distinct DOA's $\{\theta_k\}$ (with $\theta_k \neq \theta_j$ for $k \neq j$) the matrix A has full column rank:

$$\operatorname{rank}(A) = n. \tag{7.3}$$

We also assume that the signals and the noise in (7.1) are circularly symmetric Gaussian random vectors with the following first and second-order properties:

$$
\begin{aligned}
E[x(t)] &\triangleq b \neq 0 \\
E\{[x(t)-b][x(s)-b]^*\} &= P\delta_{t,s} \\
E\{[x(t)-b][x(s)-b]^T\} &= 0 \\
E\{[x(t)-b]e^*(s)\} &= 0 \\
E\{[x(t)-b]e^T(s)\} &= 0 \\
E[e(t)] &= 0 \\
E[e(t)e^*(s)] &= Q\delta_{t,s} \\
E[e(t)e^T(s)] &= 0.
\end{aligned}
\tag{7.4}
$$

The Gaussian assumption is made for the convenience of the ML approach used throughout this paper for detection and parameter estimation. However, it is not an essential assumption: without it the methodology introduced herein is no longer a ML one but it should still be statistically sound and sufficiently accurate for many other data distributions. Of the other assumptions made above the following ones are less standard: (i) The spatial covariance matrix of the noise has not been constrained in any way. In most of the previous array processing approaches Q was either assumed to be block diagonal or to satisfy a certain parametric model; and (ii) The mean of the signal vector was assumed to be different from zero. The previous discussions of the NMS scenario in the literature were either limited to the spatially white noise case [4] or to noise fields satisfying a certain spatial model [5] (as will become clear shortly postulating a noise model, as in [5], significantly complicates the signal parameter estimation problem as compared with the more general approach of this paper in which the noise is not constrained in any way!). We should also remark on the fact that the signal covariance matrix is not required to be strictly positive definite. Hence we allow for fully correlated (or coherent) signals. As explained later, an interesting feature of the proposed approach is that its (asymptotic) accuracy is not affected in any way by the degree of correlation between the signals impinging on the array.

Under the assumptions made previously all information about the signal and the noise parameters lies in the mean of the array output vector

$$E[y(t)] = Ab, \tag{7.5}$$

and in its covariance matrix

$$\Gamma \triangleq E\{[y(t) - Ab][y(t) - Ab]^*\} = APA^* + Q. \tag{7.6}$$

Furthermore, as Q has been assumed to be an arbitrary (positive definite and Hermitian) matrix, the only information on the signal parameters is provided by (7.5). Fortunately the latter information turns out to be enough for signal parameter estimation. Indeed it follows from [14][6] that the DOA's and the means of all signals impinging on the array can be uniquely determined from $E[y(t)]$ if and essentially only if:

$$b_k \neq 0, \qquad\qquad (\text{for } k = 1, \ldots, n) \tag{7.7}$$

and

$$m \geqslant 2n. \tag{7.8}$$

The "parameter identifiability" conditions above are assumed to hold throughout the remaining part of this paper. Note that condition (7.7) can be used to our advantage for focusing on certain signals of interest out of those impinging on the array. For example in a crowded environment we can "turn off" some signals by setting their means to zero. This will not only make it easier to satisfy the identifiability condition (7.8) but may also simplify the parameter estimation algorithm quite a bit (see the next sections for more details on this aspect). No similarly simple procedure for switching-off the signals that are to be ignored by the location processor appears to be available with any other existing array processing algorithm.

The main problem considered in the following sections is the estimation of the number of signals, n, and of the DOA vector, $\theta = [\theta_1 \ldots \theta_n]^T$, from the snapshots $\{y(t)\}_{t=1}^N$ obtained by the array. In the process of deriving an estimate of θ we will also obtain estimates of the less important parameters b and Γ. We make use of the ML approach for *parameter estimation* and of the generalized information criterion (GIC) rule for *signal detection*.

7.3 ML method for parameter estimation and detection

First we treat the parameter estimation problem in which we assume that n is fixed to some hypothesized value (not necessarily equal to the true one). Next we discuss the detection problem, that is the estimation of n.

7.3.1 ML parameter estimation

The negative log-likelihood function associated with the observed snapshots $\{y(t)\}_{t=1}^{N}$ is given by:

$$F(\theta, b, \Gamma) = \text{const} + N \ln |\Gamma| + \\ + \sum_{t=1}^{N} [y(t) - Ab]^* \Gamma^{-1} [y(t) - Ab], \tag{7.9}$$

where $|\Gamma|$ denotes the determinant of Γ. Minimization of (7.9) with respect to Γ, for any fixed matrix A and vector b, is immediate (see, e.g., [8]): the minimizing Γ matrix is given by

$$\tilde{\Gamma}(\theta, b) = \frac{1}{N} \sum_{t=1}^{N} [y(t) - Ab][y(t) - Ab]^*. \tag{7.10}$$

We are assuming that $N \geqslant m$ to prevent cases in which the matrix in (7.10) may be singular. The ML estimate of Γ is obtained by replacing θ and b in (7.10) with their ML estimates that are determined as described below.

Insertion of (7.10) into (7.9) yields:

$$F(\theta, b, \tilde{\Gamma}(\theta, b)) = \text{const} + N \ln |\tilde{\Gamma}(\theta, b)|. \tag{7.11}$$

Let

$$\bar{y} = \frac{1}{N} \sum_{t=1}^{N} y(t) \tag{7.12}$$

denote the sample mean of the snapshot sequence. Also let

$$\hat{R} = \frac{1}{N} \sum_{t=1}^{N} y(t) y^*(t) \tag{7.13}$$

and let

$$\hat{\Gamma} = \hat{R} - \bar{y}\bar{y}^* \tag{7.14}$$

denote the standard sample estimate of Γ. A straightforward calculation shows that:

$$
\begin{aligned}
|\tilde{\Gamma}(\theta, b)| &= |\hat{R} - \bar{y}(Ab)^* - Ab\bar{y}^* + (Ab)(Ab)^*| = \\
&= |\hat{\Gamma} + (\bar{y} - Ab)(\bar{y} - Ab)^*| = \\
&= |\hat{\Gamma}||I + \hat{\Gamma}^{-1}(\bar{y} - Ab)(\bar{y} - Ab)^*| = \\
&= |\hat{\Gamma}|[1 + (\bar{y} - Ab)^*\hat{\Gamma}^{-1}(\bar{y} - Ab)].
\end{aligned} \tag{7.15}
$$

To obtain the last equality above we have made use of the identity $|I+AB| = |I + BA|$ that holds for any conformable matrices A and B, see e.g. [8]. It readily follows from (7.15) that the function in (7.11) is minimized with respect to b, for any fixed θ, at:

$$
\tilde{b}(\theta) = (A^*\hat{\Gamma}^{-1}A)^{-1}(A^*\hat{\Gamma}^{-1}\bar{y}) \tag{7.16}
$$

and also that the corresponding minimum value of (7.11) is given by:

$$
F(\theta, \tilde{b}(\theta), \tilde{\Gamma}(\theta, \tilde{b}(\theta))) = \text{const} + N \ln[1 + f(\theta)] \tag{7.17}
$$

where

$$
\begin{aligned}
f(\theta) = \bar{y}^*[I - \hat{\Gamma}^{-1}A(A^*\hat{\Gamma}^{-1}A)^{-1}A^*]\hat{\Gamma}^{-1} \times \\
\times [I - A(A^*\hat{\Gamma}^{-1}A)^{-1}A^*\hat{\Gamma}^{-1}]\bar{y}.
\end{aligned} \tag{7.18}
$$

Let $\hat{\Gamma}^{-1/2}$ denote a square root of $\hat{\Gamma}^{-1}$ (in what follows we use the Hermitian square root for notational convenience), and let

$$
\Pi^{\perp}_{\hat{\Gamma}^{-1/2}A} \triangleq I - \hat{\Gamma}^{-1/2}A(A^*\hat{\Gamma}^{-1}A)^{-1}A^*\hat{\Gamma}^{-1/2} \tag{7.19}
$$

denote the orthogonal projector onto the null space of $A^*\hat{\Gamma}^{-1/2}$. By using this notation we can rewrite (7.18) in the following more compact form:

$$
f(\theta) = \bar{y}^*\hat{\Gamma}^{-1/2}\Pi^{\perp}_{\hat{\Gamma}^{-1/2}A}\hat{\Gamma}^{-1/2}\bar{y}. \tag{7.20}
$$

As the logarithm is a monotonic function, the ML estimate, $\hat{\theta}$, of the DOA vector is obtained by minimizing the function $f(\theta)$ in (7.20). The corresponding minimum value of the negative log-likelihood function is given by (to within an additive constant):

$$
F_{min} = N \ln[1 + f(\hat{\theta})]. \tag{7.21}
$$

For *general arrays* the minimization of $f(\theta)$ requires the use of an iterative search algorithm. It is worth noting that $f(\theta)$ has a form often encountered in ML or subspace fitting array processing problems (observe that $f(\theta)$ can be interpreted as resulting from the least-squares fitting of $\hat{\Gamma}^{-1/2}\bar{y}$ to the range space of $\hat{\Gamma}^{-1/2}A$). Consequently there are several specialized algorithms that can be used to minimize $f(\theta)$, such as the Alternating Projection algorithm [17] and the Gauss-Newton scoring algorithm [7]. The Alternating Projection algorithm has a built-in initialization scheme that may also be used to obtain initial estimates for the Gauss-Newton scoring recursion. Alternatively, whenever the array has an appropriate geometry and contains sufficiently many sensors, initial DOA estimates can be obtained by applying the MUSIC or ESPRIT algorithms with spatial smoothing to $\bar{y}$ viewed as a single "snapshot" from an "array" described by the equation $\bar{y} = A\bar{x} + \bar{e}$ (with obvious notation, see the next section).

In certain situations, in which we have control over the signal means, the computational burden associated with the minimization of (7.20) can be reduced by setting to zero some elements of the mean vector b. In particular if all elements of b but one are set to zero, the estimation of the DOA of the "signal of interest" becomes a one-dimensional problem. Switching-off certain signals may also have a beneficial effect on the accuracy with which we estimate the DOA's of the remaining signals. It is interesting to note that by setting $b_k = 0$ we switch-off the kth signal from the ML location processor. However there will still be a (zero mean) kth signal impinging on the array, which can be used for a different purpose than location. This capability of eliminating a signal from the localization processor without nulling it completely is not shared by the other approaches that do not make use of NMS.

Remark 1. *Let*

$$\tilde{f}(\theta) = tr[\Pi^{\perp}_{\hat{\Gamma}^{-1/2}A}\hat{\Gamma}^{-1/2}\hat{R}\hat{\Gamma}^{-1/2}]. \tag{7.22}$$

Observe that

$$\begin{aligned}\tilde{f}(\theta) - f(\theta) &= tr[\Pi^{\perp}_{\hat{\Gamma}^{-1/2}A}\hat{\Gamma}^{-1/2}(\hat{R} - \bar{y}\bar{y}^*)\hat{\Gamma}^{-1/2}] = \\ &= m - n.\end{aligned} \tag{7.23}$$

Hence minimizing $f(\theta)$ with respect to θ is equivalent to minimizing $\tilde{f}(\theta)$. The latter function has the form of a deterministic (or conditional) ML criterion for a model with an approximately whitened noise term. However since *the use of $f(\theta)$ for estimating θ appears to be computationally more convenient than the use of $\tilde{f}(\theta)$ we focus on $f(\theta)$ in what follows.* $\square$

For *uniform and linear arrays (ULA)* an asymptotically valid approximation to the previous ML estimate of θ can be obtained by means of a non-iterative algorithm, in the manner of the MODE approach [9]. The direction vector $a(\theta)$ associated with an ULA has a Vandermonde structure:

$$a(\theta) = [1 \, e^{i\frac{2\pi d}{\lambda}\sin(\theta)} \, \ldots \, e^{i(m-1)\frac{2\pi d}{\lambda}\sin(\theta)}]^T, \tag{7.24}$$

where d is the interelement spacing, λ is the signal wavelength, and the angle θ is measured with respect to the broadside of the array. If we let $B(z)$ denote the following polynomial

$$B(z) = \sum_{k=0}^{n} \beta_k z^k \triangleq \beta_n \prod_{k=1}^{n} [z - e^{i\frac{2\pi d}{\lambda}\sin(\theta_k)}], \tag{7.25}$$

then it is readily verified that the matrix

$$B^* = \begin{bmatrix} \beta_0 & \ldots & \beta_n & & 0 \\ & \ddots & & \ddots & \\ 0 & & \beta_0 & \ldots & \beta_n \end{bmatrix} \qquad (m-n) \times m \tag{7.26}$$

satisfies

$$B^* A = 0\,; \text{ and } rank(B) = m - n \tag{7.27}$$

(note that not all $\{\beta_k\}$ can be zero). Hence the columns of $\hat{\Gamma}^{1/2} B$ span the null space of $(\hat{\Gamma}^{-1/2} A)^*$, which implies that

$$\Pi^{\perp}_{\hat{\Gamma}^{-1/2} A} = \Pi_{\hat{\Gamma}^{1/2} B} \triangleq \hat{\Gamma}^{1/2} B (B^* \hat{\Gamma} B)^{-1} B^* \hat{\Gamma}^{1/2}. \tag{7.28}$$

Insertion of (7.28) into (7.20) yields the following reparameterized form of the ML criterion:

$$g(\beta) = \bar{y}^* B (B^* \hat{\Gamma} B)^{-1} B^* \bar{y}, \tag{7.29}$$

where $\beta = [\beta_0 \dots \beta_n]^T$ is the new parameter vector. The exact minimization of the reparameterized function still requires the use of an iterative search algorithm. However, in large samples the minimizer of (7.29) can be well approximated by using a non-iterative algorithm. Briefly stated the reason for this is that for sufficiently large values of N and at the true parameter values the vector $B^*\bar{y}$ is (stochastically) of the order of $N^{-1/2}$. By using this fact along with a standard Taylor series expansion technique ([9]) it is a relatively straightforward exercise to show that the matrix $(B^*\hat{\Gamma}B)^{-1}$ in (7.29) can be replaced by a consistent estimate without affecting the asymptotic properties of the minimizer of (7.29). Combining the above observation and (7.29) results in *the following non-iterative algorithm that yields DOA estimates with the same asymptotic properties as those of the estimates obtained by the exact minimization of the criterion in (7.29)(or (7.20))* :

(i) Obtain (consistent) initial estimates of $\{\beta_k\}$ by minimizing the function in (7.29) with $B^*\hat{\Gamma}B$ set to $\begin{bmatrix} I & 0 \end{bmatrix} \hat{\Gamma} \begin{bmatrix} I \\ 0 \end{bmatrix}$.

(ii) Obtain an asymptotic approximation of the minimizer of $g(\theta)$ by minimizing the function in (7.29) with B in $(B^*\hat{\Gamma}B)^{-1}$ built from the initial estimate derived in step (i).

Possibly run step (ii) once more by making use of the latest available estimate of β to build the matrix B in $(B^*\hat{\Gamma}B)^{-1}$.

(iii) Obtain DOA estimates from the angular positions of the zeros of the estimated $B(z)$ polynomial (cf. (7.25)).

The minimization problems in steps (i) and (ii) of the above algorithm are both quadratic and hence their solutions can readily be calculated in closed form. In the simulation study in Section V we make use of the previous non-iterative algorithm to obtain initial DOA estimates.

7.3.2 GIC Detection Rule

The number of free real-valued parameters in our ML estimation problem is equal to

$$3n + m(m+1). \tag{7.30}$$

According to the GIC principle [1] the number of signals can be estimated as the value of n that minimizes the following criterion over the set of possible positive integers

$$\begin{aligned} GIC(n) = 2N \ln[1 + f(n;\hat{\theta})] + 3nK_N, \\ (n = 0, 1, 2, \text{etc}), \end{aligned} \tag{7.31}$$

where we have stressed the dependence of $f(\hat{\theta})$ on n, and where $2N$ is the number of real-valued observations available. The factor K_N in (7.31) should be chosen such that

$$K_N \to \infty \qquad \text{and} \qquad K_N/N \to 0 \qquad \text{as } N \to \infty \tag{7.32}$$

to ensure that the GIC detection rule is consistent (i.e. it detects the true value n with a probability that goes to one as $N \to \infty$). More exactly, but still in somewhat loose terms, K_N can be chosen between $\ln(\ln(2N))$ and $\ln(2N)$ [1]. More precise rules for selecting K_N are not available in the general case. In the simulations in Section V we will use the GIC rule (7.31) with $K_N = \ln(\ln(2N))$.

7.4 Statistical analysis and the CRB

Let B be an $m \times (m - n)$ matrix the columns of which span the null space of A^*. In other words B satisfies the equation (7.27) in the previous section. Because here we address the case of a *general array* the matrix B does not necessarily have the expression in (7.26) (which is only valid for ULA's). For an array with arbitrary geometry there is in fact no closed-form expression known for B. Below we make use of B only for an intermediate reparameterization of the ML criterion; hence the lack of an explicit expression for this matrix will not introduce any problems.

By using B we can rewrite the ML criterion as in equation (7.29) (which is repeated here for easy reference):

$$g(\beta) = \bar{y}^* B(B^* \hat{\Gamma} B)^{-1} B^* \bar{y}. \tag{7.33}$$

As already explained in the previous section, a straightforward Taylor series based analysis (see, e.g., [9]) shows that replacement of $\hat{\Gamma}$ in (7.33) by the "true" covariance matrix Γ has no effect on the asymptotic properties of the minimizer of the function in (7.33). This means that the minimizers of (7.33) and of the following function have the same asymptotics:

$$\tilde{g}(\beta) = \bar{y}^* B (B^* Q B)^{-1} B^* \bar{y}. \tag{7.34}$$

To obtain (7.34) we have also used the fact that, by (7.27),

$$B^* \Gamma B = B^* Q B \tag{7.35}$$

Reverting (7.34) back to the original parameterization via θ, we arrive at the function

$$h(\theta) = \bar{y}^* Q^{-1/2} \Pi^{\perp}_{Q^{-1/2}A} Q^{-1/2} \bar{y}. \tag{7.36}$$

The vector $\bar{y}$ in (7.36) satisfies the following equation, which is easily derived from (7.1),

$$\bar{y} = A\bar{x} + \bar{e} \tag{7.37}$$

where

$$\bar{x} = \frac{1}{N}\sum_{t=1}^{N} x(t); \qquad \bar{e} = \frac{1}{N}\sum_{t=1}^{N} e(t). \tag{7.38}$$

The "noise" vector $\bar{e}$ is circularly symmetric Gaussian distributed with zero mean and the following covariance matrix,

$$E[\bar{e}\bar{e}^*] = \frac{1}{N^2}\sum_{t=1}^{N}\sum_{s=1}^{N} Q\delta_{t,s} = Q/N. \tag{7.39}$$

It follows from (7.38) and (7.39) that the transformed mean vector $Q^{-1/2}\bar{y}$ appearing in (7.36) satisfies the equation:

$$Q^{-1/2}\bar{y} = (Q^{-1/2}A)\bar{x} + Q^{-1/2}\bar{e} \tag{7.40}$$

where the "noise" is spatially white. An important consequence of the discussion above is that the criterion function in (7.36) corresponds to the deterministic ML method applied to the (fictitious) model in (7.40) (recall that Q in (7.40) is unknown). Another equally important consequence is that the *large-sample* analysis for the original array model (7.1) becomes a *high-SNR* analysis for (7.40)! (Hereafter SNR=signal-to-noise ratio). Indeed the covariance matrix of the "noise" vector in (7.40) is equal to $(1/N)I$ whereas the covariance matrix of the "signal" vector contains a term that does not depend on N:

$$E[\bar{x}\bar{x}^*] = \frac{1}{N^2}\sum_{t=1}^{N}\sum_{s=1}^{N}(P\delta_{t,s} + bb^*) = bb^* + P/N. \tag{7.41}$$

The grand conclusion of the previous discussion is that the asymptotic (in N) covariance matrix of the ML estimate of θ can be conveniently derived by applying to (7.40) the results in [11] about the high-SNR asymptotics of the deterministic ML method. (Only a slight adjustment of the analysis in [11] is required to take into account the fact that $\bar{x}$ in (7.40) is a random vector). By making use of this "shortcut derivation" we reach the conclusion that the asymptotic (for $N \gg 1$) covariance matrix of the ML estimate of θ obtained from the minimization of the criterion in (7.20) is given by (see [11]):

$$\begin{aligned} &AsCov(\hat{\theta}) = \\ &= \frac{1}{2N}\left\{Re[(D^*Q^{-1/2}\Pi^{\perp}_{Q^{-1/2}A}Q^{-1/2}D) \odot (bb^*)^T]\right\}^{-1}, \end{aligned} \tag{7.42}$$

where $\odot$ denotes elementwise matrix multiplication, $Re(\cdot)$ denotes the real part of the quantity between parentheses, and

$$D = [\frac{\partial a(\theta_1)}{\partial \theta_1} \dots \frac{\partial a(\theta_n)}{\partial \theta_n}]. \tag{7.43}$$

Next we note that the ML estimation problem dealt with in this paper is regular: the number of unknown parameters does not depend on N and in addition the ML estimates of all these parameters are readily shown to be consistent in N (i.e. these estimates tend to the true values as $N \to \infty$). As a consequence the performance of the proposed ML estimates attains the performance corresponding to the CRB, as N increases. In particular this means that:

$$CRB(\hat{\theta}) = AsCov(\hat{\theta}) \tag{7.44}$$

where $CRB(\hat{\theta})$ denotes the CRB matrix corresponding to $\hat{\theta}$.

7.5 Discussion and numerical performance study

7.5.1 Discussion

In this section we study the detection and estimation performance of the proposed ML methodology by means of Monte-Carlo simulations. We do not consider any other specific method in our performance study since no good alternative to the method proposed here seems to exist. Apparently the only general alternative to the stochastic ML approach of this paper is a deterministic ML approach in which the signals $\{x(t)\}$ are assumed to be non-random quantities. However the latter approach has several drawbacks:

- It appears to be parameter identifiable only if the noise covariance matrix Q belongs to a certain class and is parameterized in some convenient way. Hence it is not always applicable; and when it is, it requires postulating a noise model (such as a spatial AR equation) and may thus be prone to modeling errors.

  The stochastic ML method of this paper gets around the requirement of specifying a suitable model for Q by reparameterizing the likelihood function via the matrix Γ. Recovering Q from Γ (once θ has been obtained) is not possible unless Q is restricted in some way. This shows that the class of problems that are parameter identifiable with respect to all parameters $\{\theta, b, P$ and $Q\}$ indeed is not completely general. Since most often the elements of Q are nuisance parameters the aforementioned limitation is unnecessary.

- It appears that in the general case the maximization of the deterministic ML criterion must be conducted with respect to both the signal and noise parameters simultaneously. That is to say, the nuisance noise parameters cannot be concentrated out from the deterministic likelihood function. The explicit maximization of the latter function with respect to the noise parameters appears to be possible only in special cases, such as for ULA's and AR spatial noise, and even then only in an approximate manner (see, e.g., [5]).

- Finally, the asymptotics of the deterministic ML estimate of the DOA parameter vector are the same under both the deterministic and the stochastic signal assumptions (e.g. [7][9]). This means that the (asymptotic) covariance matrix corresponding to the deterministic MLM (ML method) is bounded from below by the (asymptotic) covariance matrix of the stochastic MLM proposed here, as the latter matrix equals the $CRB(\hat{\theta})$. In other words the former method can never outperform the latter one, in large samples.

7.5.2 Numerical examples

The purpose of the numerical examples is to study the performance of the proposed detection and estimation technique using simulated data. An ULA comprising $m = 10$ sensors with inter-element spacing equal to half the wavelength is used. There are two sources at 10 and 20 degrees relative to the broadside of the array. The source signals are white and Gaussian distributed with means and variances equal to 1. The noise is assumed to be spatially correlated with covariance matrix Q given by

$$[Q]_{k,l} = \rho \left(\frac{9}{10}\right)^{|k-l|} e^{i\frac{\pi}{2}(k-l)}, \tag{7.45}$$

where ρ is a scalar that is varied to adjust the SNR to the desired value. The SNR is defined as,

$$SNR = 10 \log \frac{\sigma_s}{\rho}. \tag{7.46}$$

In (7.46) σ_s is the power of the source signals and in our case we have $\sigma_s = 2$ for both sources. In the numerical examples $N = 100$ independent snapshots have been used in the calculation of the array covariance matrix and each result shown below has been obtained by averaging the outcomes of 1000 Monte Carlo simulations. The DOA estimates are obtained by using the non-iterative algorithm discussed in Section III followed by a simplex-based minimization of $f(\theta)$ in (7.18). The minimization is done using the "fminsearch" function of MATLAB initialized with the DOAs provided by the non-iterative algorithm.

In the first example we study the performance of the GIC detection rule with $K_N = \ln(\ln(2N))$. The results are shown in Figure 7.1 where the probability of detection is plotted as a function of the SNR. It is clear from the figure that the method is able to achieve a high probability of detection

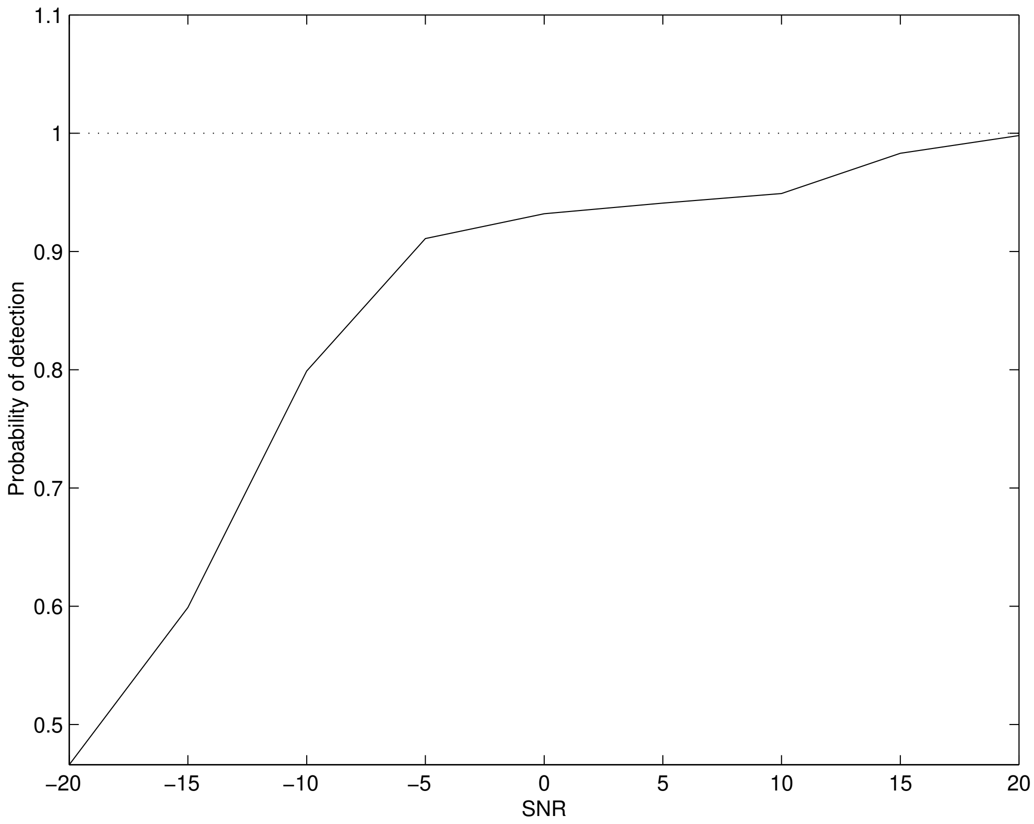

Figure 7.1: Probability of detection (solid line) as a function of the SNR for two sources located at 10 degrees and 20 degrees.

even at low SNR. Next we vary the location of one source while keeping the other source at 10 degrees. The SNR is kept at 0 dB and the results are presented in Figure 7.2 where the probability of detection is plotted against the separation between sources. As can be seen the detection probability decreases when the sources get close. When the source separation reaches zero, so does the probability of detection, which was expected.

To study the performance of the estimation algorithm, we have used the mean squared error (MSE) as a measure of performance, defined as

$$MSE(\hat{\theta}) = \frac{1}{1000} \sum_{k=1}^{1000} (\theta_{true} - \hat{\theta}^{(k)})^2, \tag{7.47}$$

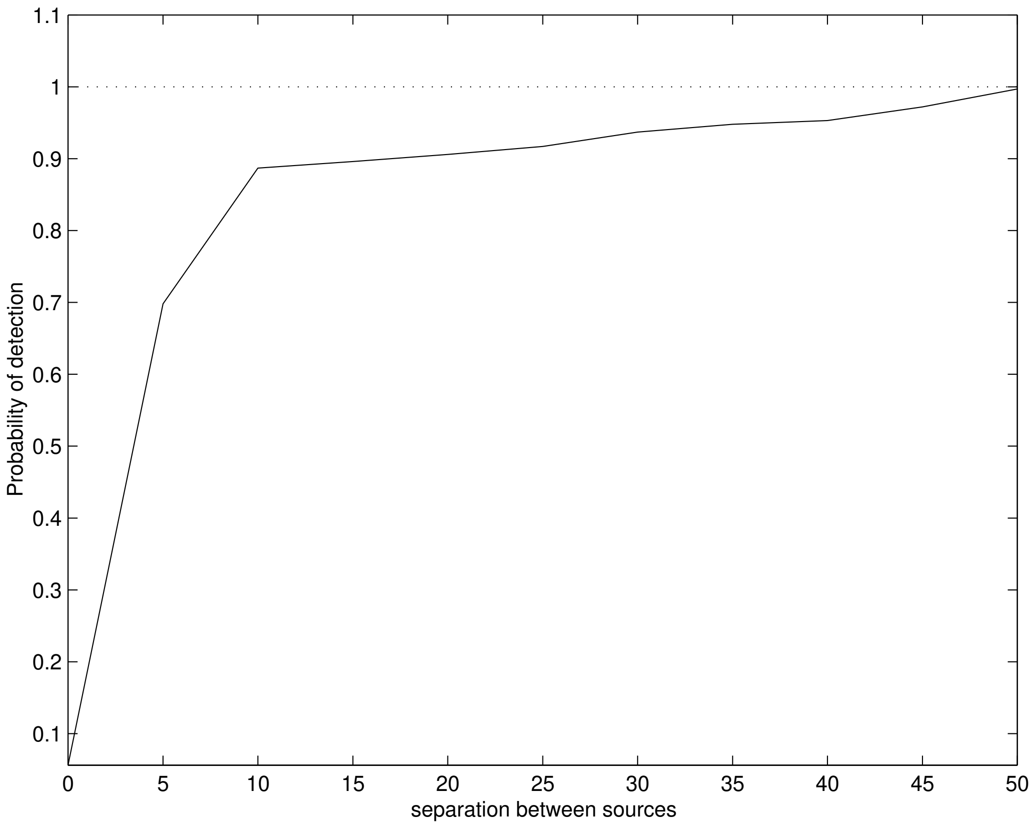

Figure 7.2: Probability of detection (solid line) as a function of the separation between sources, when one source is located at 10 degrees and the other is moved from 10 degrees to 60 degrees.

where θ_{true} is the true DOA and $\hat{\theta}^{(k)}$ is the estimated DOA in the kth realization. First we let the source signals be uncorrelated and the results are shown in Figure 7.3 where the MSE in the estimation of the DOA of the source at 10 degrees along with the corresponding CRB are plotted as a function of the SNR (the results for the source at 20 degrees were similar).

Figure 7.4 shows the same type of results as Figure 7.3 but for the case when the sources are coherent. The results are very similar to those in Figure 7.3, which is no surprise in view of the fact that the CRB does not depend on the signal covariance matrix (see 7.42)!.

Next we study the sensitivity of the method to the number of sensors

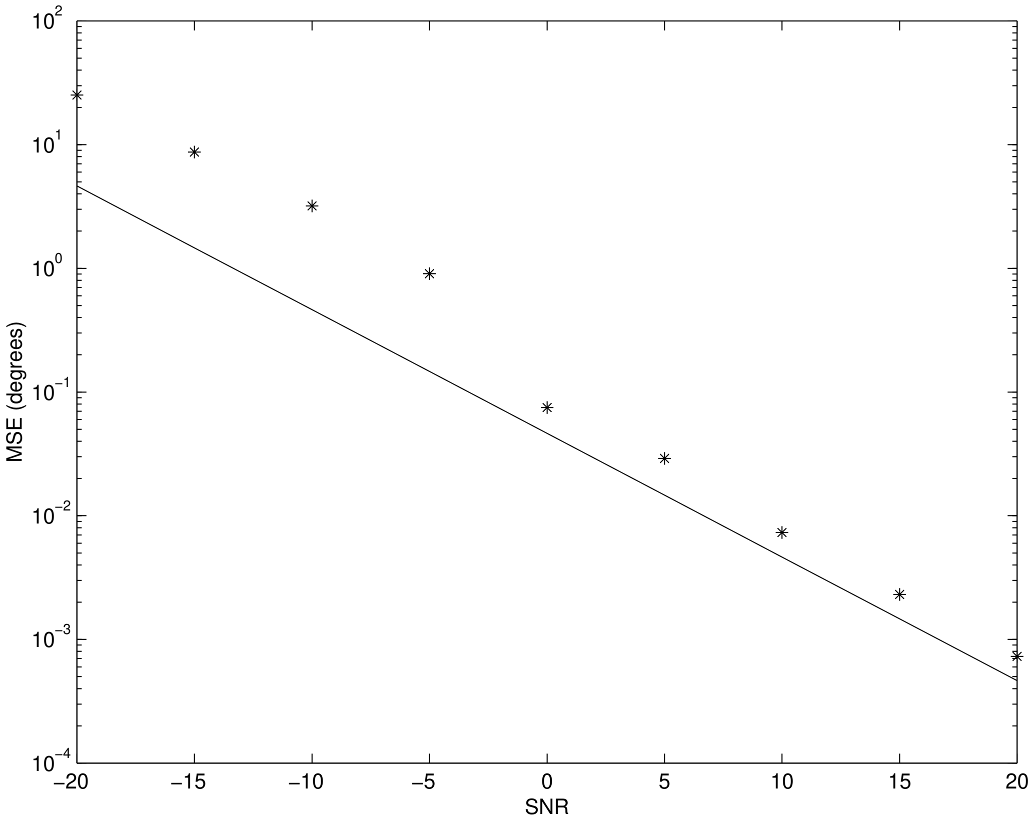

Figure 7.3: MSE of the DOA estimates for the source at 10 degrees ("*") and the CRB (solid line) as a function of the SNR, for the case of two uncorrelated sources at 10 degrees and 20 degrees.

in the array, with SNR fixed at 0 dB. In Figures 7.5 and 7.6 the MSE and CRB for the source at 10 degrees are plotted as a function of the number of sensors, for the case of correlated and uncorrelated sources. It is easily seen that both plots show the same type of behavior and that the MSE is close to the CRB.

Figures 7.7 and 7.8 show the MSE of the estimates for the source at 10 degrees and the CRB as a function of the number of snapshots, both for the case of correlated sources and the case of uncorrelated sources, respectively.

Finally we study the influence of the source separation on the estimation accuracy. One source is kept at 10 degrees and the location of the other

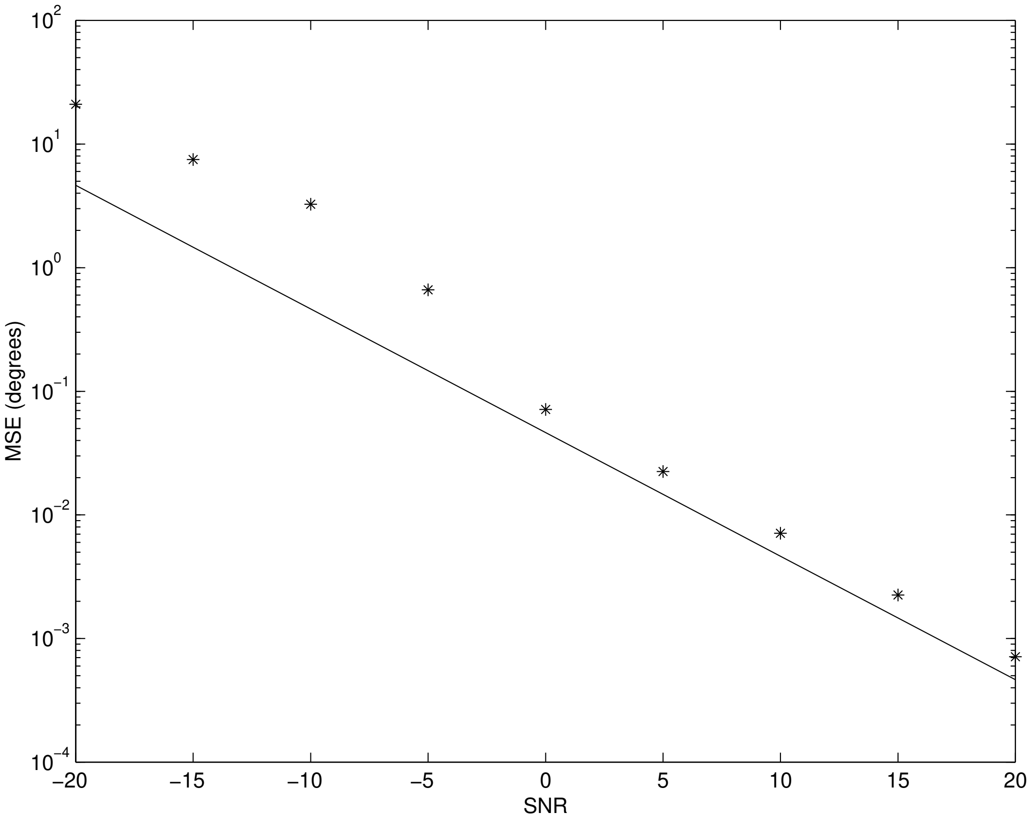

Figure 7.4: MSE of the DOA estimates for the source at 10 degrees ("*") and the CRB (solid line) as a function of the SNR, for the case of two coherent sources at 10 degrees and 20 degrees.

source is varied. The SNR is kept at 0 dB. The results are displayed in Figures 7.9 and 7.10 where the MSE for the source at 10 degrees is plotted as a function of the separation between the sources. This is done both for uncorrelated and coherent sources. As expected the estimation breaks down when the sources are close and becomes better as the separation between the sources increases.

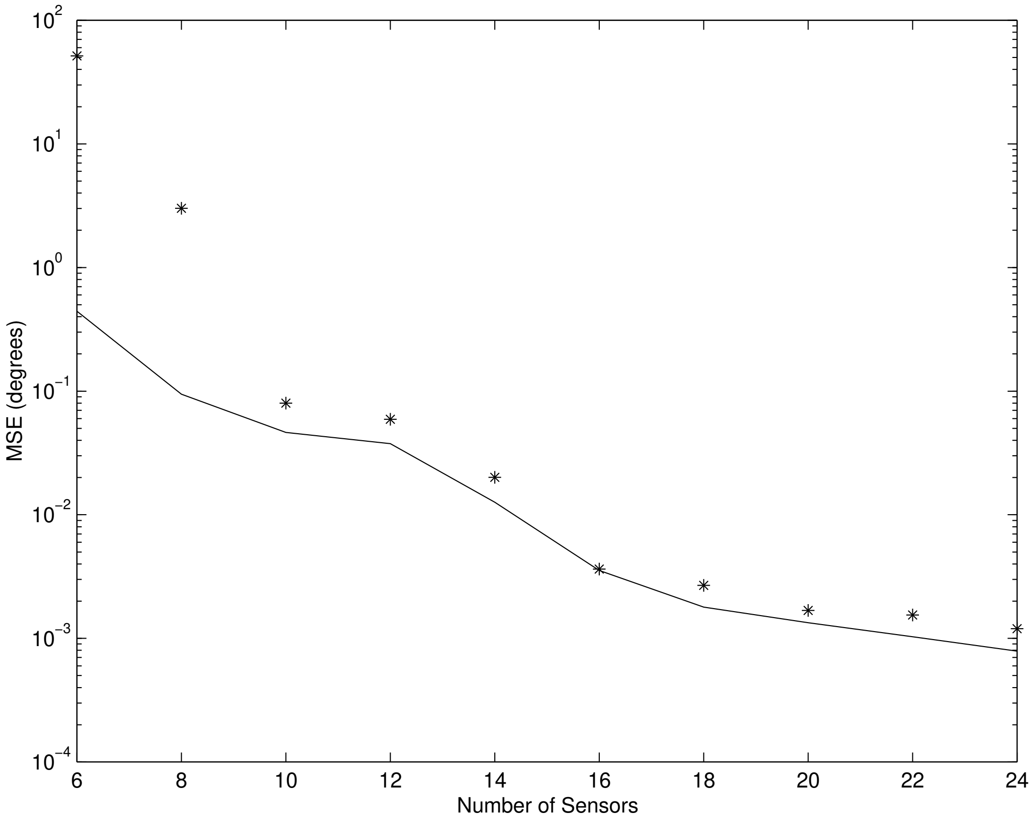

Figure 7.5: MSE of the DOA estimates for the source at 10 degrees ("*") and the CRB (solid line) as a function of the number of sensors, for the case of two uncorrelated sources at 10 degrees and 20 degrees.

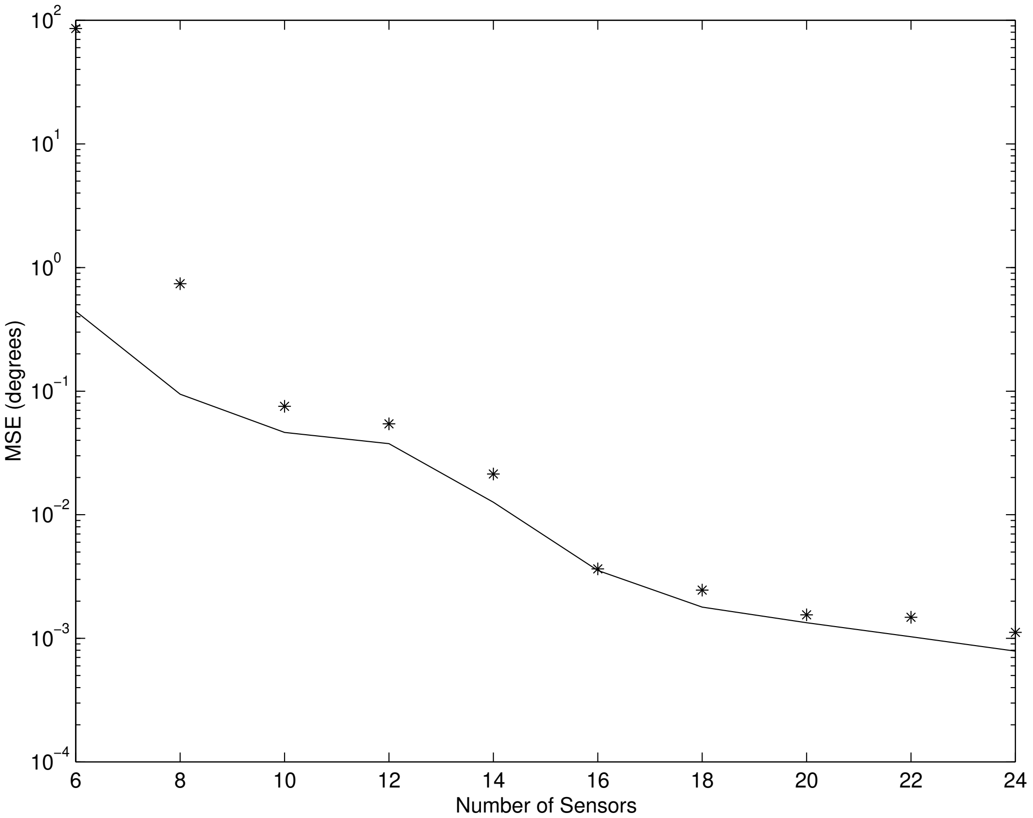

Figure 7.6: MSE of the DOA estimates for the source at 10 degrees ("*") and the CRB (solid line) as a function of the number of sensors, for the case of two coherent sources at 10 degrees and 20 degrees.

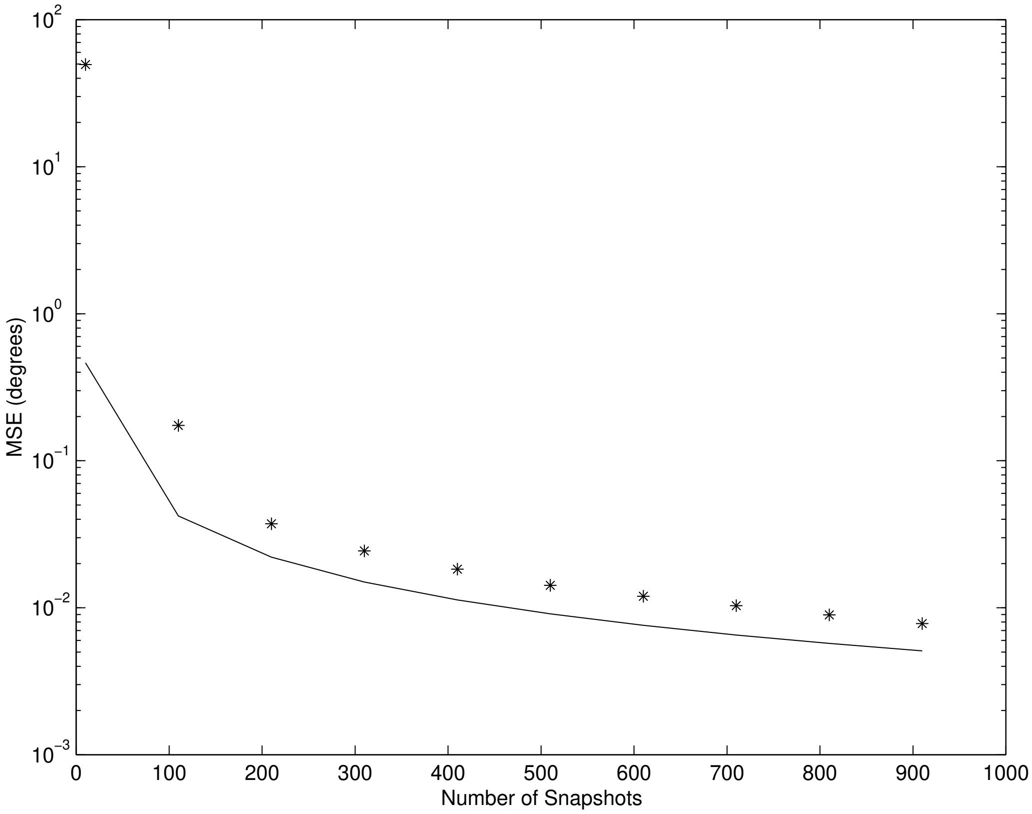

Figure 7.7: MSE of the DOA estimates for the source at 10 degrees ("*") and the CRB (solid line) as a function of the number of snapshots, for the case of two uncorrelated sources at 10 degrees and 20 degrees.

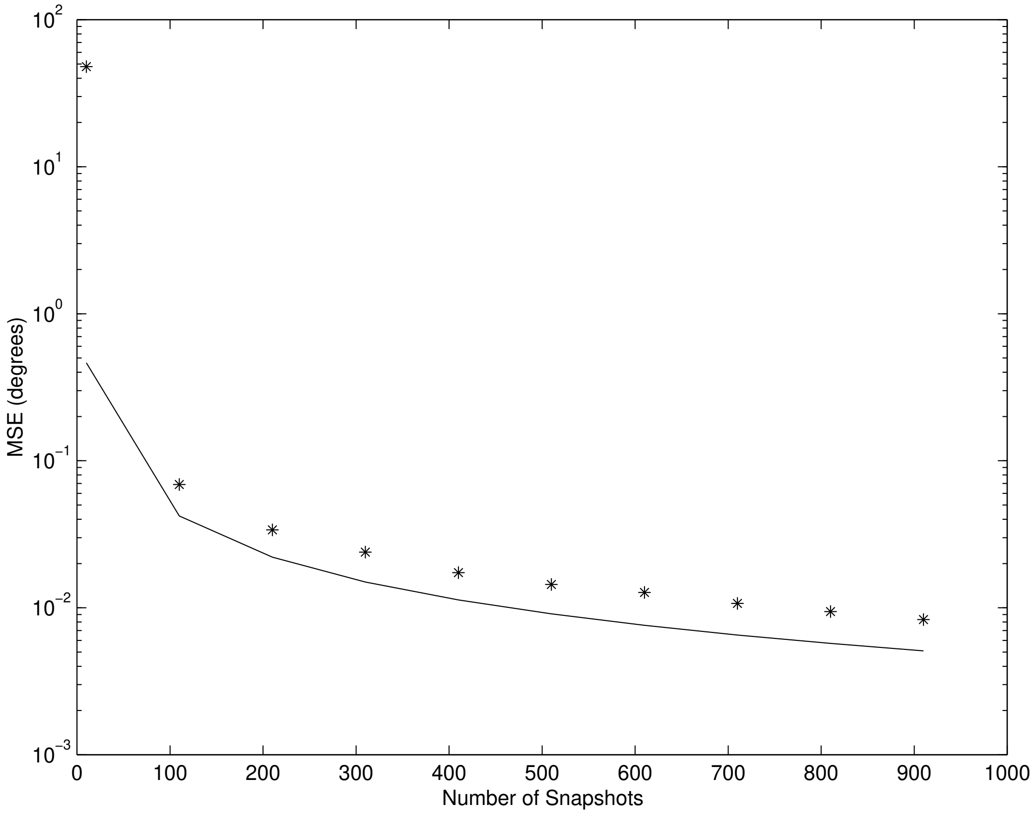

Figure 7.8: MSE of the DOA estimates for the source at 10 degrees ("*") and the CRB (solid line) as a function of the number of snapshots, for the case of two coherent sources at 10 degrees and 20 degrees.

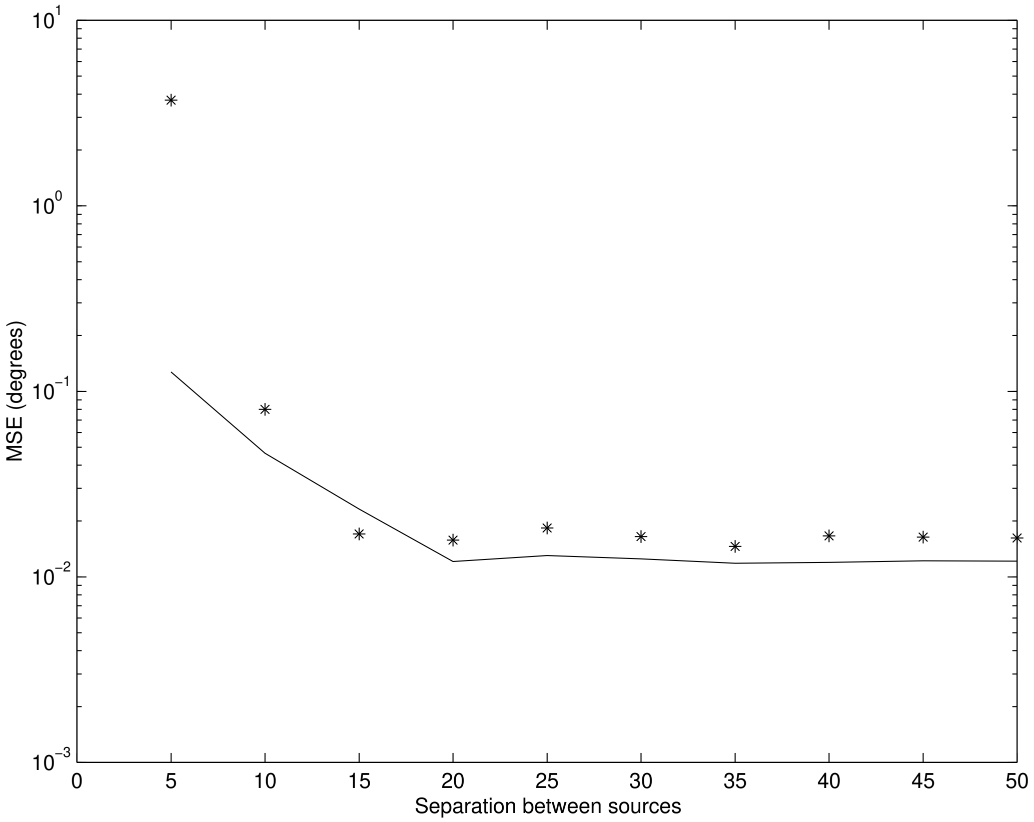

Figure 7.9: MSE of the DOA estimates for the source at 10 degrees ("*") and the CRB (solid line) as a function of the separation between two uncorrelated sources, one at 10 degrees and the other moving from 10 degrees to 60 degrees.

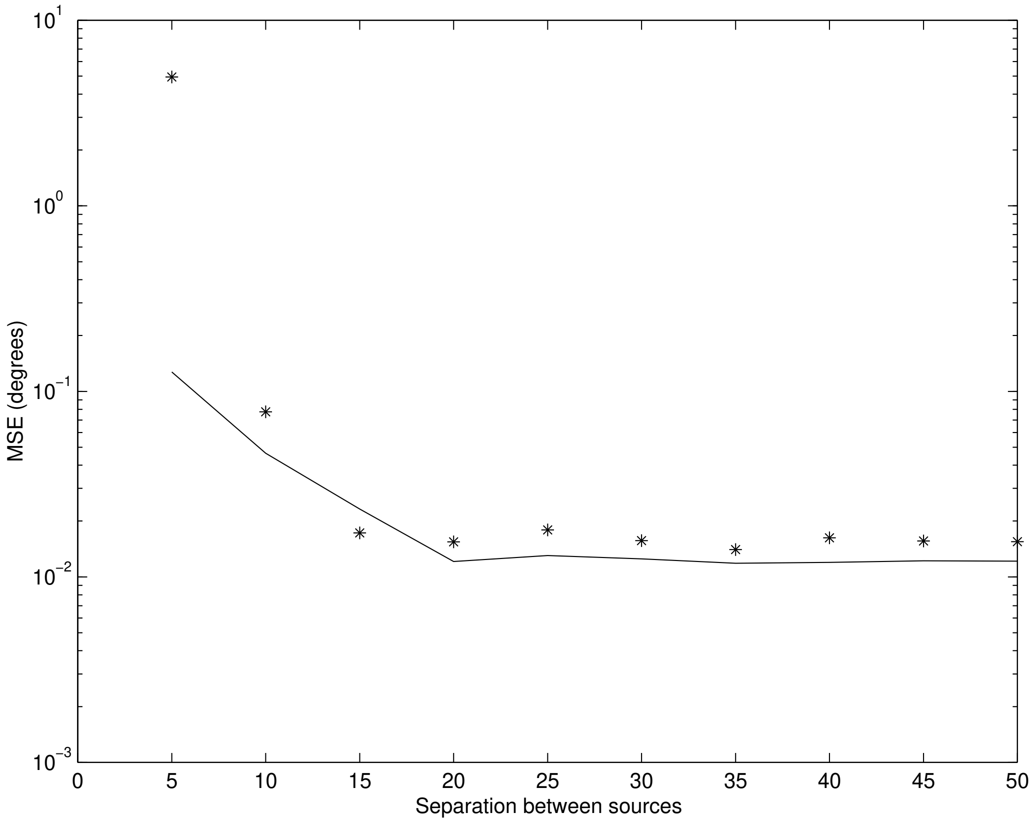

Figure 7.10: MSE of the DOA estimates for the source at 10 degrees ("*") and the CRB (solid line) as a function of the separation between two coherent sources, one at 10 degrees and the other moving from 10 degrees to 60 degrees.

Bibliography

[1] Hannan EJ, Deistler M. *The Statistical Theory of Linear Systems.* John Wiley & Sons, New York, 2nd edition, 1998.

[2] Le Cadre JP. Parametric methods for spatial signal processing in the presence of unknown colored noise fields. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37:965–983, July 1989.

[3] Li J, Halder B, Stoica P, Viberg M. Computationally efficient angle estimation for signals with known waveforms. *IEEE Transactions on Signal Processing*, 43:2154–2163, September 1995.

[4] Muravchik C, Nehorai A. Localizing and identifying underground objects using gravitational measurements. In *28th ASILOMAR Conference on Signals, Systems and Computers, Pacific Grove, CA, 1994*, Oct 1994.

[5] Nagesha V, Kay S. Maximum likelihood estimation for array processing in colored noise. *IEEE Transactions on Signal Processing*, 44:169–180, Feb 1996.

[6] Nehorai A, Starer D, Stoica P. Direction-of-arrival estimation in applications with multipath and few snapshots. *Circuits, Systems and Signal Processing*, 10:327–342, 1991.

[7] Ottersten B, Viberg M, Stoica P, Nehorai A. Exact and large sample ML techniques for parameter estimation and detection in array processing. In Haykin S, Litva J, Shepherd TJ, editor, *Radar Array Processing.* Springer-Verlag, New-York, USA, 2nd edition, 1999.

[8] Söderström T, Stoica P. *System Identification.* Prentice-Hall, International, UK, 1989.

[9] Stoica P, Sharman KC. Maximum likelihood methods for direction-of-arrival estimation. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38:1132–1143, July 1990.

[10] Stoica P, Viberg M, Ottersten B. Instrumental variable approach to array processing in spatially correlated noise fields. *IEEE Transactions on Signal Processing*, 42:121–133, Jan 1994.

[11] Viberg M. Sensitivity of parameter direction finding to colored noise fields and undermodeling. *Signal Processing*, 34:207–222, Nov 1993.

[12] Viberg M, Stoica P, Ottersten B. Array processing in correlated noise fields based on instrumental variables and subspace fitting. *IEEE Transactions on Signal Processing*, 43:1187–1199, May 1995.

[13] Viberg M, Stoica P, Ottersten B. Maximum likelihood array processing in spatially correlated noise fields using parameterized signals. *IEEE Transactions on Signal Processing*, 45:996–1004, Apr 1997.

[14] Wax M, Ziskind I. On unique localization of multiple sources by passive sensor arrays. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37:996–1000, July 1989.

[15] Wu Q, Wong KM. Estimation of DOA in unknown noise: Performance analysis of UN-MUSIC and UN-CLE, and the optimality of CCD. *IEEE Transactions on Signal Processing*, 43:454–468, Feb 1995.

[16] Ye H, DeGroat RD. Maximum likelihood DOA estimation and asymptotic Cramer-Rao bounds for additive unknown colored noise. *IEEE Transactions on Signal Processing*, 43:938–949, Apr 1995.

[17] Ziskind I, Wax M. Maximum likelihood localization of multiple sources by alternate projection. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 46:1553–1560, October 1998.

Recent licentiate theses from the Department of Information Technology

2000-011 Bharath Bhikkaji: *Model Reduction for Diffusion Systems*

2001-001 Erik Borälv: *Design and Usability in Telemedicine*

2001-002 Johan Steensland: *Domain-based partitioning for parallel SAMR applications*

2001-003 Erik K. Larsson: *On Identification of Continuous-Time Systems and Irregular Sampling*

2001-004 Bengt Eliasson: *Numerical Simulation of Kinetic Effects in Ionospheric Plasma*

2001-005 Per Carlsson: *Market and Resource Allocation Algorithms with Application to Energy Control*

2001-006 Bengt Göransson: *Usability Design: A Framework for Designing Usable Interactive Systems in Practice*

2001-007 Hans Norlander: *Parameterization of State Feedback Gains for Pole Assignment*

2001-008 Markus Bylund: *Personal Service Environments — Openness and User Control in User-Service Interaction*

2001-009 Johan Bengtsson: *Efficient Symbolic State Exploration of Timed Systems: Theory and Implementation*

2001-010 Johan Edlund: *A Parallel, Iterative Method of Moments and Physical Optics Hybrid Solver for Arbitrary Surfaces*

2001-011 Pär Samuelsson: *Modelling and control of activated sludge processes with nitrogen removal*

2001-012 Per Åhgren: *Teleconferencing, System Identification and Array Processing*

UPPSALA
UNIVERSITY

University Printers, Ekonomikum, Uppsala, Sweden