

LOCALIZATION OF PACKET BASED RADIO TRANSMITTERS IN SPACE, TIME AND FREQUENCY

BY GORAN IVKOVIC

A dissertation submitted to the
Graduate School—New Brunswick
Rutgers, The State University of New Jersey
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
Graduate Program in Electrical and Computer Engineering

Written under the direction of
Predrag Spasojevic
and approved by

New Brunswick, New Jersey

May, 2013

ABSTRACT OF THE DISSERTATION

Localization of Packet Based Radio Transmitters in Space, Time and Frequency

by Goran Ivkovic

Dissertation Director: Predrag Spasojevic

We consider the scenario where one or more sensors observe a frequency band potentially used by multiple radio transmitters forming packet based networks. Our goal is to develop algorithms for estimation of spectrum usage in space, time, and frequency. This estimation is obtained by performing some form of analysis of the received signals at the sensors. The proposed algorithms can be used for achieving efficient spectrum utilization by identifying unused portions of spectrum in space, time and frequency as well as for other applications requiring spectrum monitoring.

The received signals consist of packets from multiple transmitters with possible time-frequency collisions. Each received signal consists of multiple statistically homogeneous segments where each combination of active transmitted signals creates one or more of such segments. In order to perform any form of statistical analysis using conventional methods for stationary or cyclostationary signals these segments must be first localized in time. In the first part of the thesis we propose a nonparametric algorithm for solving this problem. Initial segmentation is computed using a variant of mean shift algorithm, which is a clustering tool based on nonparametric estimate of the underlying probability distribution. We show that this type of mean shift algorithm is based on the modified Newton's method and provide a convergence analysis which explains how and why

the algorithm works. Final segmentation results are obtained after applying a cluster validation procedure and impulse noise filtering on the initial segmentation results.

In the second part of the thesis we propose a method for analysis of the segments localized in the first step. This method is useful if transmitted signals are linearly modulated or can be approximated as sums of linearly modulated signals. For each set of segments generated by the same combination of the transmitted signals we compute a certain two dimensional slice of the fourth order spectrum. These slices are arranged in a three way array. We show that under certain conditions it is possible to recover contributions of individual signals to the observed three way array by decomposing the array into low rank terms. Thus, for each received signal we can estimate its spectrum and the associated activity sequence in time. We discuss the uniqueness conditions, treat the nontrivial problem of fourth order spectrum estimation and propose a numerical algorithm for estimation of the spectra and the associated activity sequences of individual signals from the observed three way array.

The algorithms for segmentation and fourth order spectrum based analysis require only one sensor. In the third part of the thesis we assume that multiple sensors are available. Using the algorithms mentioned above for each transmitter we can estimate its received spectrum at different sensors. From the received spectra of the same transmitted signal at different sensors it is possible to estimate the source signal spectrum and transfer functions of the channels between the transmitter and the sensors. For realistic channels the resulting deconvolution problem is often ill conditioned and then the solutions provided by existing algorithms are useless. We show that a useful solution to the formulated blind deconvolution problem can be obtained using the l_1 norm regularization and propose a numerical algorithm for estimation of the channel parameters and the source spectrum from the received spectra at the sensors. The estimated channel parameters depend on the location of the transmitter relative to the sensors and thus, they can be used for transmitter localization in space.

Table of Contents

Abstract	ii
1. Introduction	1
1.1. Problem formulation	1
1.2. Possible applications	2
1.3. Related work	3
2. Signal segmentation	8
2.1. Problem formulation	8
2.2. Segmentation algorithm	12
2.2.1. Mean shift clustering algorithm	14
2.2.2. Selection of scale parameters	19
2.2.3. Cluster validation	21
2.2.4. Removal of impulse noise	23
2.2.5. Summary of the algorithm	25
2.2.6. Fusion of different segmentation decisions	26
2.3. Numerical examples	29
2.4. Conclusion	41
3. Fourth Order Spectrum Based Analysis	43
3.1. Signal analysis method	43
3.2. Estimation of fourth order spectra	61
3.3. Computational methods	75
3.4. Numerical examples	83
3.5. Conclusion and future work	96

4. Algorithm with Multiple Sensors	98
4.1. Outline of the algorithm	98
4.2. SIMO blind deconvolution	100
4.3. Numerical example	108
4.4. Conclusion and future work	116
References	117

Chapter 1

Introduction

1.1 Problem formulation

In this thesis we consider the scenario where one or more sensors observe a frequency band potentially used by multiple radio transmitters forming packet based networks. Packet based networks consists of multiple nodes which communicate by exchanging packets using a certain protocol. Some examples of these networks are 802.11a/b/g, Bluetooth, various types of cordless phones, etc. Transmitters in these networks are active only during their packet transmission slots and silent rest of the time. This type of transmission yields signals with nonpersistent excitation. Thus, each transmitter transmits a modulated signal which can be statistically characterized by its second and higher order spectra and whose activity over time is represented with an on/off sequence. Transmissions of different nodes from the same network are coordinated by their protocol, which is typically designed to eliminate or minimize time-frequency collisions. Transmissions of nodes from different types of networks (e. g., Bluetooth and WLAN) are not coordinated in any way and may result in time-frequency collisions.

Received signal at each sensor consists of packets from different transmitters with possible time-frequency collisions between the packets coming from heterogeneous systems. Our first goal is to develop methods for analyzing the received signal at each sensor. Main goal of the single sensor analysis is the estimation of spectra and associated activity sequences of the packet based signals present. In other words, the goal of the single sensor analysis is to localize the packet based signals in time and frequency. When multiple sensors are available we also wish for each identified transmitter to estimate the signal spectrum and the channel transfer functions between the transmitter and sensors. From the estimated channel transfer function we can, at least

approximately, localize the transmitter in space. Therefore, our goal in this work is to develop algorithms for localization of packet based radio transmitters in space, time, and frequency.

1.2 Possible applications

Let us now discuss possible applications of the algorithms solving the formulated problem. By solving the formulated problem we obtain the key information about the transmitters using the observed frequency band: their spectra showing what radio spectrum frequencies they occupy, their activity sequences in time and their approximate locations in space. Thus, we obtain the complete picture of radio spectrum utilization in space, time, and frequency. Obviously, this information enables us to monitor spectrum usage in the observed frequency band, which is one possible application of these algorithms.

Another important observation is that by knowing radio spectrum utilization in space, time, and frequency we are also identifying unused portions of radio spectrum. In other words we are finding space-time-frequency spectrum holes, which are caused by static allocation of radio spectrum. Most existing wireless systems use static spectrum allocation. Since statically allocated spectrum is only partially used across space and time, this type of allocation creates space-time-frequency spectrum holes. This fact implies inefficient radio spectrum utilization. In order to achieve efficient radio spectrum utilization a new class of wireless systems with dynamic spectrum allocation is proposed [1] [2]. These new systems are supposed to exploit unused radio frequencies. By solving the formulated problem we are identifying the space-time-frequency spectrum holes, which can be exploited by the systems with dynamic spectrum allocation. Therefore, estimation of spectrum usage in space, time, and frequency provides the crucial information needed for achieving efficient spectrum utilization without causing interference to the legacy systems.

1.3 Related work

One line of work which is somewhat related to our problem is spectrum sensing for cognitive radios. In this type of research it is usually assumed that a secondary network wishes to opportunistically use the spectrum allocated to some primary user. Most of this work is motivated by the emerging 802.22 standard, where the secondary 802.22 networks should exploit available spectrum allocated to digital TV stations [3]. In order to avoid causing interference to the primary user the secondary network must be able to detect if the observed spectrum is used by the primary user. By identifying the spectrum occupied by the primary user we are also finding spectrum holes which can be used by the secondary network. This problem is usually formulated as a binary hypothesis testing. Main method for solving these problems can be classified as follows: (1) building matched filter detectors for the signals of interest or for some known signal components in these signals, (2) power spectrum based methods and (3) methods based on cyclostationarity of the signals of interest [1].

An obvious brute force approach consists in building receivers for each of the signals using the observed frequency band. This approach requires considerable signal knowledge and separate receiver for each signal format from the observed frequency band. Obviously, this approach is very complicated and impractical. It is also unnecessary since signal demodulation is not needed here. A much simpler approach is to exploit deterministic signal components (e. g., pilot carriers, preambles, etc.) that are embedded in many communication signals [1]. When these deterministic signal components exist we can build matched filter detectors for them and thus, detect presence of signals of interest. In real world even when transmitted signal contains known components, these components at the receiving side are not fully known because of effects of multipath propagation, frequency offset and unknown arrival time. Dealing with these unknown parameters complicates implementation considerably and limits performance of the method [4]. Another problem is that with this approach we need a separate detector for every signal type. In some cases it is not possible to build receivers or exploit known signal components because required signal knowledge is not available (e.

g., wireless microphones [5], cordless phones [6], etc.).

Obviously, building receivers or matched filters for known signal components is not a satisfactory solution and in some cases is not even possible. Therefore, it is desirable to develop methods that do not require detailed signal knowledge and thus, are more widely applicable. These methods are based on estimating certain statistical properties of the received signal. Two most widely used statistical features are power spectral density(PSD) and cyclostationary spectra. PSD based methods are typically based on nonparametric PSD estimates computed with certain number of frequency bins. These PSD estimates are compared with thresholds to determine occupied and available frequency bins. Various PSD based methods(they are sometimes called energy detectors) have been proposed for spectrum sensing [1] [7] [8] [9]. All these methods are based on same principles and have similar properties. Main problem with these methods is their inability to separate PSD of the signal of interest from that of interfering signals and noise. Thus, PSD based methods can determine occupied and available frequency bins, but they are not useful for signal identification. Even for determining occupancy of frequency bins setting of thresholds requires known noise levels, which can be problematic in real world conditions.

Second class of statistical methods is based on cyclostationary properties of communication signals. Many communication signals are generated by certain repetitive mechanisms and are thus, associated with certain periods(e. g., symbol period, chip period, hopping period, etc.). As a result, statistics of most communication signals are periodic functions of time. This property is in contrast with stationary signals whose statistics are independent of time. Hence, most communication signals are not stationary, but cyclostationary. Periodic fluctuations of statistics in time give rise to cyclostationary spectral features, which can be used for signal detection and identification [10]. Different signal formats are often associated with different periods which yields different cyclostationary spectral features. Hence, cyclostationary spectra exhibit certain form of signal selectivity and they are much more useful for signal identification compared to PSD [11]. Most spectrum sensing methods based on cyclostationarity(e.g., [1] [7] [12] and references in [9]) are straightforward applications or simple extensions

of the general theory presented in [10].

In addition to these main methods there are numerous methods developed for specific applications [13] [8] [5]. Most of these application specific algorithms are various heuristics based on the three main classes of methods discussed earlier. More detailed overview of these algorithms can be found in [9] and references therein.

Let us now discuss if any of these methods are applicable to our problem. We can build full receivers or detectors for known signal components if we assume that such components exist and that all needed signal knowledge is available. As already mentioned, this approach is too complicated and requires separate detector for each signal format. Therefore, this solution is not satisfactory. Methods based on estimating signal statistics, such as power spectrum or cyclostationary spectra, assume that the estimated statistical parameters do not change during the time interval over which these parameters are estimated. Since in our problem the received signals consist of multiple signals with nonpersistent excitation it is very difficult to pick the time interval for estimation during which the estimated parameters do not change. This fact prevents a straightforward application of these statistical methods in our problem.

Another important point is that most of these spectrum sensing methods explicitly or implicitly assume only one transmitter in the observed time-frequency window. If multiple transmitters are present in the observed time-frequency window these methods cannot produce a comprehensive picture of spectrum usage in space, time, and frequency. There are very few attempts to deal with multiple transmitters [14] [15]. These approaches treat some special cases and cannot be directly applied to our problem.

Obviously, the mathematical framework used in spectrum sensing research is not appropriate for our problem. A more appropriate mathematical framework for our problem can be found in blind signal separation research. A motivating example for this area of research is so-called cocktail party problem [16]. In this problem we have several speakers (or other sound sources) in the same room talking over each other. The sounds are recorded by an array of microphones placed at different locations in the room. Each microphone receives a linear combination of source signals and possibly

their attenuated and delayed versions due to reflections from surrounding objects. The goal is to recover source signals from the recorded signals at different microphones. This is an example of blind signal separation problem. The problem is formulated as estimation of source signals and impulse responses of the channels between the sources and sensors.

Since blind signal separation problems arise in many different contexts, numerous approaches have been developed for solving them [17]. These problems are solved by exploiting some properties of source signals. One class of algorithms assumes that source signals are statistically independent and non-gaussian. Source signals are recovered by finding statistically independent components from the observed signals at sensors. These methods are known as independent component analysis(ICA) algorithms [18] [19] [20]. In these approaches statistical independence is usually imposed through high order statistics [17]. Blind signal separation algorithms are not always based on statistical independence of source signals. Other approaches separate signals by assuming different autocorrelation functions(or power spectral densities) [21], different cyclostationary features [22], different time-frequency distributions [23], etc. Some approaches exploit nonstationary nature of source signals [24] [25] [26]. They are applicable to speech signals, which are highly nonstationary.

An important class of algorithms has been developed for blind signal separation problems arising in wireless communications. These algorithms exploit special properties of analog and digital communication signals such as constant modulus [27] or finite alphabet property [28] [29]. A very good overview of these methods (also known as blind beamforming algorithms) is given in [30]. They are applicable when receiver has multiple antennas. If such a receiver receives several interfering signals, individual source signals can be recovered using these blind signal separation methods. Adding multiple antennas introduces spatial dimension into signal processing algorithms. Hence, these receivers can deal with multiple superimposed signals in a way that is not possible in a single antenna receiver [31].

In our problem the received signals consist of contributions from multiple radios.

Our goal is to estimate spectra and activity sequences of the sources and impulse responses of the channels between the sources and sensors. Thus, our goal is not to estimate the source signals as in blind signal separation problems but some other parameters characterizing individual contributions of the different sources. Another important point is that most blind signal separation methods assume that the number of sensors is higher than the number of sources, which is unrealistic in most practical applications. There are some attempts to solve underdetermined blind signal separation problems [32] [33], but these solutions require additional assumptions and are not satisfactory. In our problem, we wish to design algorithms which work even if only one sensor is available. Therefore, our parameter estimation problem cannot be solved by straightforward application of the existing blind signal separation algorithms and it requires a different approach.

Chapter 2

Signal segmentation

In this chapter we consider a scenario where one or more sensing nodes observe a frequency band used by multiple packet based transmitters. In this environment the received signals at the sensors consists of multiple statistically homogeneous segments. Each combination of active transmitted signals and sensor noise create one or more of such segments with their distinct statistics. Our goal is to develop a segmentation algorithm which localizes in time these signal segments. First we will consider a single sensor case and then we extend our approach to the scenario with multiple sensors.

2.1 Problem formulation

We consider a setup where one sensing node observes a frequency band with M packet based radio sources. Received complex baseband equivalent signal at the sensing node is

$$r(t) = \sum_{m=1}^M \sum_{l=0}^{L_c} a_{m,l} s_m(t - \tau_{m,l}) + n(t) \quad (2.1)$$

where $s_m(t)$ are transmitted signals, $a_{m,l}$ and $\tau_{m,l}$ are channel parameters, and $n(t)$ is stationary sensor noise. Transmitted signals are given by

$$s_m(t) = \sum_{q=-\infty}^{\infty} g_m(t; \boldsymbol{\theta}_{m,q}) (u(t - t_{in,m,q}) - u(t - t_{fi,m,q})) \quad (2.2)$$

where $u(t)$ is a unit step function, $t_{in,m,q}$ and $t_{fi,m,q}$ denote start and stop time of the q -th packet, and $g_m(t; \boldsymbol{\theta}_{m,q})$ defines modulation format of the m -th source. In general, $g_m(t; \boldsymbol{\theta}_{m,q})$ depends on the parameter vector $\boldsymbol{\theta}_{m,q}$ containing certain parameters that may vary from one packet to another. For example, for linear modulation formats $g_m(t; \boldsymbol{\theta}_m) = \sum_{k=-\infty}^{\infty} a_k p_m(t - kT_m - \phi_m)$ where a_k are information symbols, $p_m(t)$ is the pulse shape, T_m is the symbol period, and $\boldsymbol{\theta}_m = [\phi_m]$ is the timing parameter that

may vary across packets. We assume that during the observation interval each source operates with a constant power when on.

We would like obtain information on signals present in the observed frequency band through some form of statistical analysis of the received signal. Let us now review main existing statistical methods for signal analysis and see how they can be used here. There is a well established theory for analysis of stationary signals. Methods for analysis of stationary signals using second order statistics are well known [34] [35]. There are also methods for analysis of stationary signals using higher order statistics [36]. These methods can be extended to cyclostationary signals, where statistics periodically vary in time. We also have second order [10] and higher order methods [37] [38] for analysis of cyclostationary signals. Methods for analysis of stationary signals assume that statistics of the analyzed signal are independent of time on the observation time interval. Methods for analysis of cyclostationary signals assume that statistics of the analyzed signal periodically vary in time but parameters describing this periodic variation are constant on the observation time interval. Therefore, these main statistical methods for analysis of stationary and cyclostationary signals are applicable to signals that are in certain sense statistically homogeneous on the observation time interval. These methods applicable to stationary and cyclostationary signals do not extend easily to general nonstationary signals [39]. Hence, there is no general theory applicable to nonstationary signals and they are treated on a case by case basis. For some special classes of nonstationary signals it is possible and useful to define certain statistics, which have different interpretations and properties from those of the similar statistics of stationary and cyclostationary signals [40]. However, it is not always possible or useful to define such statistics for nonstationary signals.

We see that in our case the received signal $r(t)$ is nonstationary consisting of a number of statistically homogeneous segments. Each combination of active transmitted signals and noise generates one or more such segments. Statistical parameters abruptly change at time instants located at the boundaries between the segments. These time instants are called statistical change points. In order to use any of the conventional statistical methods we have to localize in time the statistically homogeneous segments

of the received signal. One possible approach is to use methods for detection of statistical change points. A very good overview of these methods can be found in [41] and numerous references therein. Many of these methods are based on likelihood ratio tests and they have some nice theoretical properties. However, they are derived assuming a certain probability distribution for the underlying signals. Since these probability distributions are almost never known in practical applications, these methods must resort to simplifying assumptions (e. g., Gaussian signals) which are rarely satisfied for any real world signals. In addition to the assumption on the signals probability distributions the methods for detection of statistical change points frequently assume various parametric models for the underlying signals (e. g., AR, ARMA, state space models, etc.). These parametric models impose implicit assumptions on the analyzed signals and hence, they are not always appropriate. Since these parametric methods for statistical change detection are based on unrealistic assumptions a number of nonparametric methods for statistical change detection and signal segmentation have been developed. An on-line method for detecting change points was proposed in [42]. This method is based on tracking signal subspace of the estimated signal covariance matrix. This method is applicable if the underlying signals have a finite dimensional signal subspace which is not necessarily true for all signals. It is assumed in [42] that the dimension of the signal subspace is known in advance and same for all segments. However, in most applications the dimension of the signal subspace is not known and it must be estimated which is a very difficult problem itself. In [43] the signal to be segmented is first transformed into another sequence which is constructed so that the statistical changes in the original signal result in the changes of the mean value of the transformed signal. Then the segmentation is performed by detecting the changes of the mean value of the transformed signal. Main weakness of this approach is that the transformed sequence cannot be a vector sequence which is more appropriate for some applications [44]. In [45] the signal to be segmented is transformed into a sequence of vectors where the change points of the original signal produce changes of the mean values of the transformed vector sequence. The change points are detected on the transformed vectors using an

on-line approach based on a certain distance measure which is constructed using Gaussian kernel function. Main problem with this method is that it works properly only if the scale parameters of the Gaussian kernel function are appropriately selected which is a difficult problem without a good solution. Another nonparametric approach based on numerical estimation of likelihood ratio was proposed in [46]. The likelihood ratio is estimated using a linear combination of Gaussian kernel functions which is similar to kernel probability density estimates [47]. Main problem with this approach is that this type of estimation works well only if the scale parameters of the Gaussian kernels are appropriately selected. In [46] these parameters are determined using some heuristic method. However, it is very difficult to find a good solution for this problem since the scale parameters depend on the likelihood ratio function which is unknown and has to be estimated. In conclusion, each of these nonparametric methods has significant limitations and some unresolved problems and hence, none of them is quite satisfactory.

In our problem we wish to localize in time statistically homogeneous segments where the signal in each segment consists of the sensor noise and possibly some combination of the received communication signals. These communication signals may be generated using different modulation formats and thus, they may have different probability distributions. Therefore, the signals from different segments of the received signal typically have different probability distributions which are unknown a priori. Clearly, finding appropriate parametric models for these signals is very difficult and hence, parametric methods for statistical change detection do not represent a satisfactory solution to our problem. In this paper we present a nonparametric approach. We analyze the underlying signal by sliding a window of a certain length over the signal and computing a feature vector in every step. The transformation for computing the feature vectors is chosen so that statistical changes in the analyzed signal are transformed into the changes of the mean value of the feature vectors. The feature vector used here is the signal power spectrum computed with a certain frequency resolution. Now, the problem of signal segmentation is reduced to the problem of clustering of the feature vectors. Each cluster contains the feature vectors from one or more segments containing the same combination of active transmitted signals. The clustering problem is solved in

two steps. An initial clustering is obtained using a variant of the mean shift algorithm with adaptive scale parameters, which is a clustering method based on nonparametric estimate of the probability density function of the feature vectors. The initial clustering is refined in the cluster validation step where our goal is to remove all the clusters caused by spurious peaks in the pdf estimate and keep only the true clusters. The solution of the clustering problem produces initial segmentation results which sometimes contain some amount of impulse noise. This noise is removed using one form of generalized vector median filtering, which yields the final segmentation results. Our approach is completely nonparametric and it does not assume any probability distributions or parametric models for the underlying signals.

2.2 Segmentation algorithm

Let the total observed bandwidth be W . The received signal $r(t)$ is bandlimited to $[-W/2, W/2]$ by filtering and sampled with $T_s \leq 1/W$, which yields the discrete time signal $r(k)$, $k = 1, \dots, K$, where KT_s is the total observed time. We analyze the signal $r(k)$ with the sliding window of the length N_T , which corresponds to time resolution $T = N_T T_s$. We define the vectors

$$\mathbf{r}_n = \begin{bmatrix} r(1 + (n-1)N_d) & \dots & r((n-1)N_d + N_T) \end{bmatrix}^T \quad (2.3)$$

for $n = 1, \dots, N$ where $1 \leq N_d \leq N_T$ is step parameter of the sliding window. We compute the feature vectors $\mathbf{x}_n = f(\mathbf{r}_n)$ for $n = 1, \dots, N$ where the function $f(\cdot)$ should be chosen so that each set of segments generated by the same combination of received signals and noise produces a subset or cluster of the vectors \mathbf{x}_n with its distinct mean vector. In other words, the function $f(\cdot)$ should transform the problem of finding statistically homogeneous segments in the received signal $r(k)$ into clustering of the vectors \mathbf{x}_n for $n = 1, \dots, N$. Here, we assume that each combination of received signals and noise creates a signal which is uniquely identified by its power spectrum and use power spectrum estimates of the vectors \mathbf{r}_n as the feature vectors \mathbf{x}_n . If this is not the case other appropriate signal features (e. g., other signal spectra, projections onto a suitable set of basis functions, time-frequency distributions, etc.) can be used instead

of the power spectrum. If parametric signal models (e. g., AR, ARMA, state space models, etc.) are appropriate then the coefficients of these models can be used as the feature vectors and thus, any apriori knowledge of the analyzed signal can be taken into account. We will explain later how different types of feature vectors computed from the same signal can be used simultaneously for segmentation.

For each of the vectors \mathbf{r}_n , we compute its power spectrum estimate as follows. We compute the vectors

$$\mathbf{v}_n = FFT\{\mathbf{r}_n\} \quad (2.4)$$

for $n = 1, \dots, N$. Next, we compute $\mathbf{v}_n \circ \mathbf{v}_n^*$ for $n = 1, \dots, N$ where the symbol \circ denotes elementwise (Hadamard) product of two vectors [48]. The vector $\mathbf{v}_n \circ \mathbf{v}_n^*$ contains magnitude squared entries of \mathbf{v}_n . We compute the power spectrum estimate

$$\mathbf{x}_n = \mathbf{A}[\mathbf{v}_n \circ \mathbf{v}_n^*] \quad (2.5)$$

for $n = 1, \dots, N$ where the frequency smoothing matrix

$$\mathbf{A} = \begin{bmatrix} 1 & \dots & 1 & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\ 0 & \dots & 0 & 1 & \dots & 1 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 0 & \dots & 1 & \dots & 1 \end{bmatrix} \quad (2.6)$$

is of the size J by N_T and its each row has $L = \lceil N_T/J \rceil$ consecutive ones. The power spectrum estimate \mathbf{x}_n is one form of a periodogram with frequency smoothing [34]. The vector \mathbf{x}_n has J frequency bins where $1 \leq J \leq N_T$.

For appropriately selected N_T the vectors \mathbf{x}_n for $n = 1, \dots, N$ form a certain number of clusters, where each cluster contains the subset of vectors \mathbf{x}_n whose analysis windows of the length N_T are from the segments with the same combination of active sources. In general, each cluster corresponds to a union of one or more continuous time segments. Dispersion around cluster means is caused by two effects: (1) inevitable power spectrum estimation errors due to the finite sample size effects and (2) presence of vectors \mathbf{x}_n corresponding to time intervals with transitions from one segment to another. Reducing power spectrum estimation errors requires a larger N_T , while reducing the number of

time intervals with transitions requires a smaller N_T . Therefore, any choice of N_T is a compromise between these two conflicting requirements. When N_T is very small (i. e., close to one) it becomes impossible to reliably estimate power spectrum or any other statistics because of large estimation errors. When N_T is too large the percentage of time intervals with transitions is high, which leads to merging of clusters. Hence, the vectors \mathbf{x}_n form clusters only for N_T selected from a certain useful range. Our fundamental assumption is that duration of the statistically homogeneous segments in $r(k)$ is such that an appropriate range for selection of N_T exist. Under this assumption, finding statistically homogeneous segments in $r(k)$ reduces to finding clusters in the vectors \mathbf{x}_n .

2.2.1 Mean shift clustering algorithm

One clustering approach that is useful for our problem assumes that the vectors \mathbf{x}_n are realizations from a certain multimodal pdf. The idea is to somehow determine cluster assignments from an estimate of this pdf obtained from \mathbf{x}_n . There are two main approaches to do this: parametric and nonparametric. Parametric methods assume some functional form for the underlying pdf and then estimate its parameters. Most frequently assumed functional form is the Gaussian mixture model where each cluster is modeled with one multivariate Gaussian distribution [49]. Nonparametric methods do not impose any assumed model on the pdf. In our application the shape the underlying pdf depends on the dispersion effects caused by the PSD estimation errors and by the vectors \mathbf{x}_n corresponding to time intervals with transitions from one segment to another. The pdf depends on the analyzed signals and the parameters N_T , J , and L . Since the pdf is signal dependent and unknown, it is not appropriate to impose any parametric models here. For this reason we proceed using nonparametric methods.

Our clustering approach is based on the kernel pdf estimate [47]. Given realizations \mathbf{x}_n for $n = 1, \dots, N$ the nonparametric pdf estimate with Gaussian kernel is

$$\hat{f}(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^N \frac{1}{(2\pi)^{J/2} \sqrt{\det \mathbf{H}_n}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{x}_n)^T \mathbf{H}_n^{-1}(\mathbf{x}-\mathbf{x}_n)} \quad (2.7)$$

where the symmetric positive definite matrix \mathbf{H}_n contains scale parameters, which

determine shape and size of the Gaussian kernel centered at realization \mathbf{x}_n . In the following analysis we will assume that the matrices \mathbf{H}_n are given. We will discuss how to determine these parameters later. Clustering algorithms using nonparametric pdf estimates are based on the assumption that each cluster contributes one mode (i. e., local maximum) to the underlying pdf. Hence, we need to assign each vector \mathbf{x}_n to its nearest mode. This assignment can be performed using the mean shift algorithm which is an iterative procedure which starts at the vector \mathbf{x}_n and converges to the nearest mode of the pdf estimate $\hat{f}(\mathbf{x})$. This idea was introduced in [50] and further studied in [51] [52] [53] [54]. Here, we will use one variant of the mean shift algorithm. We need an iterative procedure that starts at each \mathbf{x}_n and converges to the nearest mode of $\hat{f}(\mathbf{x})$. We derive the desired iterative procedure starting from Newton's method

$$\mathbf{y}_{j+1} = \mathbf{y}_j - \alpha \left[\nabla^2 \hat{f}(\mathbf{y}_j) \right]^{-1} \nabla \hat{f}(\mathbf{y}_j) \quad (2.8)$$

where \mathbf{y}_j is the iteration vector at the j -th step and $\mathbf{y}_0 = \mathbf{x}_n$ and α is the step size. The gradient of the kernel pdf estimate (2.7) is

$$\nabla \hat{f}(\mathbf{x}) = \sum_{n=1}^N \omega_n(\mathbf{x}) \mathbf{H}_n^{-1} (\mathbf{x}_n - \mathbf{x}) \quad (2.9)$$

where $\omega_n(\mathbf{x}) = C_n e^{-\frac{1}{2}(\mathbf{x}-\mathbf{x}_n)^T \mathbf{H}_n^{-1} (\mathbf{x}-\mathbf{x}_n)}$ and $C_n = \left(N(2\pi)^{J/2} \sqrt{\det \mathbf{H}_n} \right)^{-1}$. We can rewrite (2.9) as

$$\mathbf{H}(\mathbf{x}) \nabla \hat{f}(\mathbf{x}) = \mathbf{H}(\mathbf{x}) \sum_{n=1}^N \omega_n(\mathbf{x}) \mathbf{H}_n^{-1} \mathbf{x}_n - \mathbf{x} \quad (2.10)$$

where $\mathbf{H}(\mathbf{x})^{-1} = \sum_{n=1}^N \omega_n(\mathbf{x}) \mathbf{H}_n^{-1}$. The Hessian of $\hat{f}(\mathbf{x})$ is

$$\nabla^2 \hat{f}(\mathbf{x}) = \sum_{n=1}^N \omega_n(\mathbf{x}) \mathbf{H}_n^{-1} (\mathbf{x}_n - \mathbf{x}) (\mathbf{x}_n - \mathbf{x})^T \mathbf{H}_n^{-1} - \sum_{n=1}^N \omega_n(\mathbf{x}) \mathbf{H}_n^{-1} \quad (2.11)$$

where the first term is positive definite

$$\left[\nabla^2 \hat{f}(\mathbf{x}) \right]_{pd} = \sum_{n=1}^N \omega_n(\mathbf{x}) \mathbf{H}_n^{-1} (\mathbf{x}_n - \mathbf{x}) (\mathbf{x}_n - \mathbf{x})^T \mathbf{H}_n^{-1} \quad (2.12)$$

and the second term is negative definite and can be written as

$$\left[\nabla^2 \hat{f}(\mathbf{x}) \right]_{nd} = - \sum_{n=1}^N \omega_n(\mathbf{x}) \mathbf{H}_n^{-1} = -\mathbf{H}(\mathbf{x})^{-1}. \quad (2.13)$$

The iterative procedure (2.8) converges to the nearest local maximum of $\hat{f}(\mathbf{x})$ if the Hessian $\nabla^2 \hat{f}(\mathbf{y}_j)$ is negative definite. We see from (2.11) that the Hessian is, in general, indefinite. Therefore, we cannot use the ordinary Newton's method as defined in (2.8) because that would not lead to a convergent algorithm. One strategy for dealing with indefinite Hessian is to replace the true Hessian with a positive definite or negative definite matrix which is closest to the true Hessian in a certain sense. This strategy is called the modified Newton's method [55, sec. 4.4.2]. Since we are looking for a nearest local maximum of $\hat{f}(\mathbf{x})$ we replace the true Hessian $\nabla^2 \hat{f}(\mathbf{x})$ with its negative definite part $\left[\nabla^2 \hat{f}(\mathbf{x})\right]_{nd}$ and obtain the iterative procedure

$$\mathbf{y}_{j+1} = \mathbf{y}_j - \alpha \left[\nabla^2 \hat{f}(\mathbf{y}_j)\right]_{nd}^{-1} \nabla \hat{f}(\mathbf{y}_j) = \mathbf{y}_j + \alpha \mathbf{H}(\mathbf{y}_j) \nabla \hat{f}(\mathbf{y}_j) \quad (2.14)$$

which is an example of the modified Newton's method. Since $\left[\nabla^2 \hat{f}(\mathbf{x})\right]_{nd}$ is negative definite it is easy to show that the search direction $\left[\nabla^2 \hat{f}(\mathbf{y}_j)\right]_{nd}^{-1} \nabla \hat{f}(\mathbf{y}_j)$ is a direction of increase of $\hat{f}(\mathbf{x})$ [56, ch. 9]. Therefore, it is possible to select the step size α so that the algorithm (2.14) converges. To analyze convergence of the algorithm we form

$$\hat{f}(\mathbf{y}_{j+1}) - \hat{f}(\mathbf{y}_j) = \sum_{n=1}^N C_n [e^{-\frac{u_n}{2}} - e^{-\frac{v_n}{2}}] \quad (2.15)$$

where $v_n = (\mathbf{y}_j - \mathbf{x}_n)^T \mathbf{H}_n^{-1} (\mathbf{y}_j - \mathbf{x}_n)$ and $u_n = (\mathbf{y}_{j+1} - \mathbf{x}_n)^T \mathbf{H}_n^{-1} (\mathbf{y}_{j+1} - \mathbf{x}_n)$. Since the function $g(z) = e^{-z/2}$ is convex we have $g(z_2) - g(z_1) \geq g'(z_1)(z_2 - z_1)$. Applying this inequality to (2.16) yields

$$\hat{f}(\mathbf{y}_{j+1}) - \hat{f}(\mathbf{y}_j) \geq \frac{1}{2} \sum_{n=1}^N C_n e^{-\frac{v_n}{2}} [v_n - u_n]. \quad (2.16)$$

Since $C_n e^{-\frac{v_n}{2}} = \omega_n(\mathbf{y}_j)$ and $\mathbf{H}(\mathbf{y}_j)^{-1} = \sum_{n=1}^N \omega_n(\mathbf{y}_j) \mathbf{H}_n^{-1}$ we have

$$\begin{aligned} & \sum_{n=1}^N \frac{C_n}{2} e^{-\frac{v_n}{2}} [v_n - u_n] = \\ & = \frac{1}{2} \left[\mathbf{y}_j^T \mathbf{H}(\mathbf{y}_j)^{-1} \mathbf{y}_j - \mathbf{y}_{j+1}^T \mathbf{H}(\mathbf{y}_j)^{-1} \mathbf{y}_{j+1} + 2(\mathbf{y}_{j+1} - \mathbf{y}_j)^T \sum_{n=1}^N \omega_n(\mathbf{y}_j) \mathbf{H}_n^{-1} \mathbf{x}_n \right]. \end{aligned} \quad (2.17)$$

Using (2.10) and (2.14) we have

$$\sum_{n=1}^N \omega_n(\mathbf{y}_j) \mathbf{H}_n^{-1} \mathbf{x}_n = \nabla \hat{f}(\mathbf{y}_j) + \mathbf{H}(\mathbf{y}_j)^{-1} \mathbf{y}_j = \mathbf{H}(\mathbf{y}_j)^{-1} \left(\frac{1}{\alpha} (\mathbf{y}_{j+1} - \mathbf{y}_j) + \mathbf{y}_j \right). \quad (2.18)$$

After substituting (2.18) into (2.17) and some straightforward manipulations we get

$$\sum_{n=1}^N \frac{C_n}{2} e^{-\frac{v_n}{2}} [v_n - u_n] = \frac{2-\alpha}{2\alpha} (\mathbf{y}_{j+1} - \mathbf{y}_j)^T \mathbf{H}(\mathbf{y}_j)^{-1} (\mathbf{y}_{j+1} - \mathbf{y}_j). \quad (2.19)$$

From (2.14), (2.16), and (2.19) we get

$$\hat{f}(\mathbf{y}_{j+1}) - \hat{f}(\mathbf{y}_j) \geq \frac{\alpha(2-\alpha)}{2} (\nabla \hat{f}(\mathbf{y}_j))^T \mathbf{H}(\mathbf{y}_j) (\nabla \hat{f}(\mathbf{y}_j)). \quad (2.20)$$

Since $\mathbf{H}(\mathbf{y}_j)^{-1} = \sum_{n=1}^N \omega_n(\mathbf{y}_j) \mathbf{H}_n^{-1}$ where $\omega_n(\mathbf{y}_j) > 0$ and \mathbf{H}_n are symmetric positive definite matrices, we conclude that $\mathbf{H}(\mathbf{y}_j)$ and its inverse are also symmetric positive definite matrices. Therefore, the convergence of (2.14) is assured for $0 < \alpha < 2$. We pick $\alpha = 1$ since this choice maximizes the lower bound in (2.20). For $\alpha = 1$ equation (2.14) becomes

$$\mathbf{y}_{j+1} = \left[\sum_{n=1}^N \omega_n(\mathbf{y}_j) \mathbf{H}_n^{-1} \right]^{-1} \sum_{n=1}^N \omega_n(\mathbf{y}_j) \mathbf{H}_n^{-1} \mathbf{x}_n \quad (2.21)$$

which corresponds to one version of the mean shift iterations [53] [54]. We can draw the following conclusions: (1) if $\nabla \hat{f}(\mathbf{y}_j) \neq \mathbf{0}$ it follows from (2.20) that $\hat{f}(\mathbf{y}_{j+1}) > \hat{f}(\mathbf{y}_j)$ since $\mathbf{H}(\mathbf{y}_j)$ is positive definite and (2) if $\nabla \hat{f}(\mathbf{y}_j) = \mathbf{0}$ it follows from (2.14) that $\mathbf{y}_{j+1} = \mathbf{y}_j$. If $\mathbf{y}_0 = \mathbf{x}_n$ and $\nabla \hat{f}(\mathbf{y}_0) \neq \mathbf{0}$ the iterations (2.21) move the vector \mathbf{y}_j in the direction of increase of $\hat{f}(\mathbf{y}_j)$ until it reaches a point where $\nabla \hat{f}(\mathbf{y}_j) = \mathbf{0}$. Since the convergence is also assured for $0 < \alpha < 2$ the function $\hat{f}(\mathbf{y}_j)$ is monotonically increasing on the continuous trajectory obtained by connecting vectors \mathbf{y}_j for $j = 0, 1, 2, \dots$ with straight lines. Thus, when the vector \mathbf{x}_n gets attracted by one of the points of convergence it converges to that point. Depending on the shape of the underlying probability distribution estimate each point of convergence attracts a subset of the vectors \mathbf{x}_n for $n = 1, \dots, N$. All the vectors \mathbf{x}_n that converged to the same point are assigned to the same cluster. Thus, the number of clusters is equal to the number of the convergence points, which is automatically determined by the algorithm. In practice the convergence points are almost always local maxima (i.e. modes of the probability distribution estimate). Although it very rarely happens in practice, in theory it is possible that some points of convergence are not local maxima of $\hat{f}(\mathbf{x})$. These points can be detected by checking the Hessian $\nabla^2 \hat{f}(\mathbf{x})$ at the points of convergence. If the

Hessian $\nabla^2 \hat{f}(\mathbf{x})$ is not negative definite at a point of convergence, that point is not a local maximum of $\hat{f}(\mathbf{x})$. The vectors \mathbf{x}_n that converged to those points can be classified based on other vectors that did converged to the local maxima of $\hat{f}(\mathbf{x})$ using the k nearest neighbors rule [52].

There is another way to arrive at the mean shift iterations (2.21) without assuming the iterative form (2.14). Let us assume that \mathbf{y}_j is given and we wish to select \mathbf{y}_{j+1} so that the LHS in (2.16) is maximized. From (2.21) we see that this is equivalent to

$$\min_{\mathbf{y}_{j+1}} \frac{1}{2} \mathbf{y}_{j+1}^T \mathbf{H}(\mathbf{y}_j)^{-1} \mathbf{y}_{j+1} - \mathbf{y}_{j+1}^T \sum_{n=1}^N \omega_n(\mathbf{y}_j) \mathbf{H}_n^{-1} \mathbf{x}_n. \quad (2.22)$$

Since the matrix $\mathbf{H}(\mathbf{y}_j)$ is positive definite we only need to find the gradient of the expression in (2.22) and set it to zero. We get

$$\mathbf{H}(\mathbf{y}_j)^{-1} \mathbf{y}_{j+1} - \sum_{n=1}^N \omega_n(\mathbf{y}_j) \mathbf{H}_n^{-1} \mathbf{x}_n = \mathbf{0}. \quad (2.23)$$

which is equivalent to (2.21). It follows that for a given \mathbf{y}_j the vector \mathbf{y}_{j+1} given by (2.21) maximizes the LHS in (2.16).

The iterations (2.21) define the mean shift algorithm with an adaptive (variable) scale, since each vector \mathbf{x}_n is assigned its own scale parameters \mathbf{H}_n . This type of mean shift algorithm appears in [53] and [54]. However, the algorithm is not fully understood. It is not clearly explained in [53] or [54] how one can arrive at iterations (2.21). Convergence of the fixed scale algorithm was proved in [52]. Convergence of the variable scale algorithm is proved under some assumptions in [54]. However, these proofs assume $\alpha = 1$ and study convergence of the discrete sequence of points \mathbf{y}_j for $j = 0, 1, 2, \dots$. These proofs do not explain what is happening between these discrete points and hence, they cannot explain why each vector converges to the mode whose region of attraction contains that vector, which is crucial for clustering. Here, we explained how one can logically arrive at the iterations (2.21). We have shown that the mean shift algorithm with variable scale is in fact the modified Newton's method where the true Hessian $\nabla^2 \hat{f}(\mathbf{x})$ is replaced with its negative definite part $\left[\nabla^2 \hat{f}(\mathbf{x}) \right]_{nd}$ and with the step size $\alpha = 1$. In vicinity of the convergence points the algorithm behaves similarly to true Newton's method, which explains its rapid convergence. Our

convergence analysis is qualitatively different because it allows the step size α to be continuous. We have shown that $\hat{f}(\mathbf{x})$ is monotonically increasing on the continuous trajectory obtained by connecting the discrete points \mathbf{y}_j for $j = 0, 1, 2, \dots$ with straight lines. This fact explains why each vector \mathbf{x}_n converges to the local maximum of $\hat{f}(\mathbf{x})$ whose region of attraction contains the vector \mathbf{x}_n . Therefore, our derivation of the iterations (2.21) and convergence analysis go beyond existing work and demonstrate how and why the variable scale mean shift algorithm works.

2.2.2 Selection of scale parameters

Main difficulty in applying the mean shift algorithm to practical problems is in determining the appropriate scale parameters \mathbf{H}_n . To explain the role of the scale parameters we must understand their effect on the pdf estimate (2.7). Let us assume that \mathbf{H}_n are known constants independent of \mathbf{x}_n . Since \mathbf{H}_n is positive definite we can represent it as $\mathbf{H}_n = h_n^2 \mathbf{A}_n \mathbf{A}_n^T$ where $\det \mathbf{H}_n = h_n^{2J}$ and $\det \mathbf{A}_n = 1$. Then using an analysis similar to the one from [47, sec. 6.3.2] we have

$$E[\hat{f}(\mathbf{x})] = \frac{1}{N} \sum_{n=1}^N \int K(\mathbf{z}) f(\mathbf{x} - h_n \mathbf{A}_n \mathbf{z}) d\mathbf{z} \quad (2.24)$$

and

$$\text{Var}[\hat{f}(\mathbf{x})] \approx \frac{f(\mathbf{x}) \int K(\mathbf{z})^2 d\mathbf{z}}{N^2} \sum_{n=1}^N \frac{1}{h_n^J} \quad (2.25)$$

where $K(\mathbf{z}) = (1/(2\pi)^{J/2}) e^{-\frac{1}{2}\mathbf{z}^T \mathbf{z}}$ and the approximation in (2.25) is valid for small h_n . When $h_n = 0$ the estimate $\hat{f}(\mathbf{x})$ is unbiased but its variance tends to infinity. For $h_n = 0$ the estimate $\hat{f}(\mathbf{x})$ given by (2.7) consists of delta impulses and it is useless for our clustering application. In order to get a useful estimate we have to work with $h_n > 0$, which introduces some bias but reduces the variance of $\hat{f}(\mathbf{x})$. Scale parameters should be large enough so that only local maxima of $\hat{f}(\mathbf{x})$ are those due to the actual clusters in the vectors \mathbf{x}_n but not too large because then two adjacent modes could be merged. This is why the problem of scale parameter selection is very difficult.

The simplest choice is to set $\mathbf{H}_n = h^2 \mathbf{I}$ and then determine only one global scale parameter h as proposed in [52]. We experimented with this fixed scale algorithm and

were unable to obtain satisfactory results. Since the underlying pdf typically contains both peaks and flat regions it may be impossible to find one fixed scale parameter h that yields correct clustering. Next possibility is to set $\mathbf{H}_n = h_n^2 \mathbf{I}$ and then determine the parameters h_n [57]. This method is again not satisfactory, since in our application different entries of the vectors \mathbf{x}_n representing different frequency bins also require different scale parameters. Therefore, in this application we must use mean shift algorithm with adaptive scale, where each vector \mathbf{x}_n has its own scale parameters \mathbf{H}_n where different entries of \mathbf{x}_n may have different scale parameters. We select the scale parameters \mathbf{H}_n as follows. We form the matrix \mathbf{X}_n whose columns are \mathbf{x}_n and its k nearest neighbors. The scale parameters are found as

$$\mathbf{H}_n = \text{diag} \left\{ \begin{bmatrix} h_{1n} & \dots & h_{J,n} \end{bmatrix} \right\} \quad (2.26)$$

where

$$h_{j,n} = [\max(\mathbf{X}_n(j, :)) - \min(\mathbf{X}_n(j, :))]^2 \quad (2.27)$$

where $\mathbf{X}_n(j, :)$ is the j -th row of \mathbf{X}_n . If points around \mathbf{x}_n are densely packed scale parameters will be small. On the other hand, if \mathbf{x}_n lies in the region of low density scale parameters will be large. Thus, the scale parameters given by (2.27) adapt to the local properties of the underlying pdf. We picked diagonal form for \mathbf{H}_n because this is the simplest form where each entry of the vector \mathbf{x}_n can have its own scale parameter. This choice also reduces computational complexity of mean shift iterations (2.21). Since (2.21) requires computing matrix inversion, using full matrix \mathbf{H}_n would result in a much more computationally costly algorithm. According to our choice $h_n^J = \sqrt{\det \mathbf{H}_n} = h_{1,n} h_{2,n} \dots h_{J,n}$, which is the volume of the smallest J -dimensional box that contains \mathbf{x}_n and its k nearest neighbors. For any finite N these scale parameters depend on \mathbf{x}_n , which makes the analysis of bias and variance of $\hat{f}(\mathbf{x})$ very complicated. Let us assume that N is very large and that the number of nearest neighbors k depends on N so that $\lim_{N \rightarrow \infty} k(N) = \infty$ and $\lim_{N \rightarrow \infty} k(N)/N = 0$. Then the parameters h_n tends to zero as $N \rightarrow \infty$ regardless of the underlying pdf [58, sec. 6.2.]. Then we can treat h_n as small constants independent of \mathbf{x}_n and hence, we can use bias and variance expressions given by (2.24) and (2.25). Since $h_n \rightarrow 0$ as $N \rightarrow \infty$ we see from (2.24) that

the estimate $\hat{f}(\mathbf{x})$ is asymptotically unbiased. Now we note that $k(N)/(Nh_n^J)$ is the k nearest neighbors pdf estimate [58, sec. 6.2.]. Since this pdf estimate is asymptotically consistent we have $f(\mathbf{x}_n) \approx k(N)/(Nh_n^J)$. Inserting this into (2.25) we have

$$Var[\hat{f}(\mathbf{x})] \approx \frac{f(\mathbf{x}) \int K(\mathbf{z})^2 d\mathbf{z}}{Nk(N)} \sum_{n=1}^N f(\mathbf{x}_n) \leq \frac{f(\mathbf{x}) \int K(\mathbf{z})^2 d\mathbf{z}}{k(N)} \max_{\mathbf{x}} f(\mathbf{x}). \quad (2.28)$$

Assuming that $f(\mathbf{x})$ is bounded the last expression tends to zero as $N \rightarrow \infty$ since $\lim_{N \rightarrow \infty} k(N) = \infty$. Hence, when the scale parameters are given by (2.26) and (2.27) and $k(N)$ satisfies $\lim_{N \rightarrow \infty} k(N) = \infty$ and $\lim_{N \rightarrow \infty} k(N)/N = 0$ we get asymptotically unbiased and consistent estimates of $f(\mathbf{x})$. Numerical experiments have also confirmed this conclusion. For any finite N (2.26) and (2.27) yield scale parameters which adapt to the local properties of the underlying pdf. We can satisfy the conditions on $k(N)$ by setting $k(N) = N^p$ where $0 < p < 1$. Using p we can control bias variance trade-off in estimating $f(\mathbf{x})$. Increasing p from 0 to 1 increases the bias and reduces the variance.

Performance of the mean shift algorithm with the proposed selection of scale parameters is much better compared to the fixed scale algorithm. However, there are still occasional problems caused by spurious peaks in the pdf estimate. In such cases, the algorithm finds two or more closely spaced modes where only one mode exists in the true pdf and thus creates undesirable cluster splitting. Probability of cluster splitting can be reduced by increasing the parameter p in $k(N) = N^p$, but this parameter cannot be increased too much because then some actual clusters could be missed. Some other methods for selecting scale parameters were proposed in [53] [57] and [54], but as acknowledged in [54] this problem does not have a satisfactory solution. Main difficulty in selection of scale parameters is that appropriate values of scale parameters depend on the underlying pdf. However, this pdf is almost always unknown in practical applications.

2.2.3 Cluster validation

Instead of trying to find a better method for selection of scale parameters we take a different approach. Using the mean shift algorithm with the proposed selection of scale parameters we obtain an initial clustering of the vectors \mathbf{x}_n . In this initial clustering

the number of clusters may be overestimated because of the spurious peaks in the pdf estimate $\hat{f}(\mathbf{x})$. In order to remove the clusters caused by the spurious peaks of $\hat{f}(\mathbf{x})$ and find the actual clusters in the vectors \mathbf{x}_n we need a cluster validation algorithm. Some methods for solving this problem can be found in [59] [60] [61] [62]. Main idea behind these methods is to select only statistically significant modes of $\hat{f}(\mathbf{x})$, which usually means that the probability mass concentrated around those modes must be sufficiently large. Most of the existing methods treat the case where $\hat{f}(\mathbf{x})$ is a function of one variable and do not generalize well to the multidimensional case. Also performance of these methods depends heavily on the setting of various thresholds which is highly nontrivial in real world problems. Another challenging problem occurs when the estimate $\hat{f}(\mathbf{x})$ has several spurious peaks located in the vicinity of one true mode of $f(\mathbf{x})$. In this case a meaningful algorithm must recognize this situation and merge artificial clusters corresponding to the spurious peaks into one true cluster. The existing methods do not have a satisfactory solution for this problem.

Our cluster validation algorithm is based on the following distance measure between two clusters

$$d_{ij} = \frac{(\mathbf{m}_i - \mathbf{m}_j)^T (\mathbf{m}_i - \mathbf{m}_j)}{\sqrt{(\mathbf{m}_i - \mathbf{m}_j)^T \mathbf{\Sigma}_i (\mathbf{m}_i - \mathbf{m}_j)} + \sqrt{(\mathbf{m}_i - \mathbf{m}_j)^T \mathbf{\Sigma}_j (\mathbf{m}_i - \mathbf{m}_j)}} \quad (2.29)$$

where \mathbf{m}_i and $\mathbf{\Sigma}_i$ are the mean and covariance matrix of the i -th cluster. The distance measure (2.29) can be rewritten as

$$d_{ij} = \frac{\|\mathbf{m}_i - \mathbf{m}_j\|_2}{\sqrt{\frac{(\mathbf{m}_i - \mathbf{m}_j)^T \mathbf{\Sigma}_i (\mathbf{m}_i - \mathbf{m}_j)}{\|\mathbf{m}_i - \mathbf{m}_j\|_2^2}} + \sqrt{\frac{(\mathbf{m}_i - \mathbf{m}_j)^T \mathbf{\Sigma}_j (\mathbf{m}_i - \mathbf{m}_j)}{\|\mathbf{m}_i - \mathbf{m}_j\|_2^2}}} \quad (2.30)$$

where $\|\mathbf{m}_i - \mathbf{m}_j\|_2$ is the Euclidean distance between centroids of the clusters i and j and $\sqrt{\frac{(\mathbf{m}_i - \mathbf{m}_j)^T \mathbf{\Sigma}_i (\mathbf{m}_i - \mathbf{m}_j)}{\|\mathbf{m}_i - \mathbf{m}_j\|_2^2}}$ is standard deviation of the i -th cluster in direction of the unit vector $(\mathbf{m}_i - \mathbf{m}_j)/\|\mathbf{m}_i - \mathbf{m}_j\|_2$. These facts explain why it makes sense to use the quantity defined in (2.29) as a cluster distance measure. This cluster distance measure is of the similar form as reciprocal of the cluster similarity measure from [63], but the terms in the denominator in (2.29) are different from those in [63].

Our cluster validation algorithm takes as its input the number of clusters I_{cl} and the I_{cl} by N cluster assignment matrix $\mathbf{U} = [u_{qn}]$ where $u_{qn} = 1$ if \mathbf{x}_n belongs to the q -th

cluster and $u_{qn} = 0$ otherwise. Initial values for I_{cl} and \mathbf{U} are obtained by applying the mean shift iterations (2.21) on \mathbf{x}_n for $n = 1, \dots, N$. In the initial step we compute the mean vectors and covariance matrices of the clusters using

$$\mathbf{m}_q = \frac{1}{N_q} \sum_{n=1}^N u_{qn} \mathbf{x}_n \quad (2.31)$$

and

$$\mathbf{\Sigma}_q = \frac{1}{N_q} \sum_{n=1}^N u_{qn} (\mathbf{x}_n - \mathbf{m}_q)(\mathbf{x}_n - \mathbf{m}_q)^T \quad (2.32)$$

where N_q is the number of vectors in the q -th cluster and $q = 1, \dots, I_{cl}$. Next, we repeat the following two steps.

(1) Compute d_{qp} using (2.29) for $1 \leq q < p \leq I_{cl}$ and find the pair of clusters (q_m, p_m) with the smallest distance. If $d_{q_m p_m} > d_{tr}$ where d_{tr} is a predefined threshold, exit the algorithm, otherwise go to step (2).

(2) Merge the clusters q_m and p_m by updating the matrix \mathbf{U} . The q_m -th and p_m -th rows of the old matrix are added and they become one row in the updated matrix. All other rows of the old matrix \mathbf{U} are copied into the updated matrix without any changes. Set $I_{cl} = I_{cl} - 1$, compute the mean vector and covariance matrix for the new cluster resulted from merging the clusters q_m and p_m and go to step (1).

These two steps are repeated until all pairwise distances are greater than the threshold d_{tr} or the number of clusters I_{cl} is reduced to 1. The choice of d_{tr} is a compromise between two conflicting requirements: ability to detect two closely spaced true clusters and ability to eliminate false clusters resulting from spurious peaks in the pdf estimate. We see that our cluster distance measure defined in (2.29) is independent of the scaling of the vectors \mathbf{x}_n . Hence, the setting of d_{tr} is also independent of this scaling.

2.2.4 Removal of impulse noise

Applying mean shift iterations (2.21) and our cluster validation algorithm on \mathbf{x}_n for $n = 1, \dots, N$ determines the number of clusters I_{cl} and the cluster assignment matrix $\mathbf{U} = [u_{jn}] = \begin{bmatrix} \mathbf{u}_1 & \dots & \mathbf{u}_N \end{bmatrix}$. If \mathbf{x}_n belongs to the q -th cluster then $u_{qn} = 1$ and $u_{jn} = 0$ for $j \neq q$. As the final result of our algorithm we wish to find the matrix of segmentation sequences \mathbf{S} whose size is I_{cl} by K . The q -th row of \mathbf{S} has ones at time

instants when the received signal has the distinct statistics corresponding to the q -th cluster and zeros otherwise. First, we obtain the initial estimate of the segmentation matrix denoted by $\hat{\mathbf{S}}_{cl} = \begin{bmatrix} \hat{\mathbf{s}}_{cl,1} & \dots & \hat{\mathbf{s}}_{cl,K} \end{bmatrix}$. The columns of this matrix are computed as

$$\hat{\mathbf{s}}_{cl,1+(n-1)N_d+N_T/2+r} = \mathbf{u}_n \quad (2.33)$$

where $r = 1 - N_d/2, \dots, N_d/2$ and $n = 1, \dots, N$. We assumed that N_T and N_d are even numbers. Our clustering algorithm assigns each vector \mathbf{x}_n independently of all other vectors. As a result $\hat{\mathbf{S}}_{cl}$ often contains certain amount of impulse noise, especially around transition points. Therefore, we need a method for removal of impulse noise.

We use the following filtering procedure for removal of impulse noise. Let $\hat{\mathbf{S}} = \begin{bmatrix} \hat{\mathbf{s}}_1 & \dots & \hat{\mathbf{s}}_K \end{bmatrix}$ be the filtered version of segmentation sequences. The k -th column of $\hat{\mathbf{S}}$ is computed as

$$\hat{\mathbf{s}}_k = \text{majority vector}\{\hat{\mathbf{s}}_{cl,k-N_w}, \dots, \hat{\mathbf{s}}_{cl,k+N_w}\} \quad (2.34)$$

where $k = 1, \dots, K$. In other words, $\hat{\mathbf{s}}_k$ is the vector from the set $\{\hat{\mathbf{s}}_{cl,k-N_w}, \dots, \hat{\mathbf{s}}_{cl,k+N_w}\}$ which has the highest frequency of appearance. This vector can be selected because $\hat{\mathbf{S}}_{cl}$ contains only I_{cl} distinct columns. This filtering method can be considered as one version of the generalized vector median filtering [64]. This type of filtering removes short impulses but does not blur abrupt transitions in the segmentation matrix. Parameter N_w controls duration of the impulses that are removed by the filtering procedure. When N_w is small the filtering removes only very short impulses. Increasing N_w leads to removal of impulses of greater duration as well. Most vector median filters have the following property. If the filtering operation is applied repeatedly on the input signal the output eventually becomes invariant to filtering. The resulting signal is called a root signal for the given filter [64]. This method is used to compute $\hat{\mathbf{S}}$ from $\hat{\mathbf{S}}_{cl}$. The final estimate $\hat{\mathbf{S}}$ is obtained by repeatedly applying the filtering defined by (2.34) on the previous filtering output until there is no change in the output.

2.2.5 Summary of the algorithm

Let us now summarize the proposed segmentation algorithm. It consists of the following four steps.

(1) Collect the samples of the received signal $r(k)$ for $k = 1, \dots, K$. Apply the sliding window analysis with the selected parameters N_T and N_d to create the vectors \mathbf{r}_n for $n = 1, \dots, N$. Estimate PSD for each of the vectors \mathbf{r}_n to obtain the vectors \mathbf{x}_n for $n = 1, \dots, N$.

(2) Using the proposed mean shift algorithm perform clustering of the vectors \mathbf{x}_n for $n = 1, \dots, N$. This procedure gives initial estimates of the number of clusters I_{cl} and cluster assignment matrix \mathbf{U} .

(3) Apply the proposed cluster validation procedure with the selected parameter d_{tr} the initial estimates for I_{cl} and \mathbf{U} from the previous step. This procedure gives the final estimates for I_{cl} and \mathbf{U} .

(4) Find the initial segmentation matrix $\hat{\mathbf{S}}_{cl}$ using the matrix \mathbf{U} from the previous step. Apply the proposed impulse noise removal procedure with the selected parameter N_w on $\hat{\mathbf{S}}_{cl}$ which yields the segmentation matrix $\hat{\mathbf{S}}$.

In the second step we must apply the mean shift iterations (2.21) on each of the vectors \mathbf{x}_n for $n = 1, \dots, N$. When the number of vectors N is large this step can be very computationally costly since we must solve an optimization problem for each vector \mathbf{x}_n . One simple method for reduction of the computational complexity is proposed in [65]. This method consists in using the mean shift algorithm to cluster only one representative subset of the vectors \mathbf{x}_n . All other vectors are classified using the vectors from the subset and k nearest neighbors rule. In our application this subset can be generated using the vector resulting from the sliding window analysis with no overlap (i. e., $N_d = N_T$). Hence, this technique for reduction of computational complexity can be easily applied here. However, this technique has a significant drawback. Here the clustering assignments are based on a subset of vectors. Hence, in this case the estimate of the underlying pdf and the resulting cluster assignments are less accurate compared to the estimate obtained using all available vectors.

Another useful technique for speeding up the computations in the proposed algorithm is parallelization. This technique can be applied to (1) the k nearest neighbors search in the selection of scale parameters needed for the mean shift algorithm, (2) the mean shift iterations and (3) the impulse noise removal using vector median filtering. All these operations are applied to sequences of vectors (\mathbf{x}_n for $n = 1, \dots, N$ or the columns of \mathbf{S}_d). These operations can be done independently at each vector and hence, they can be also done in parallel.

2.2.6 Fusion of different segmentation decisions

By applying the proposed algorithm on the same signal with different time resolutions, different numbers of frequency bins or different types of feature vectors we can obtain different segmentation matrices of the same signal. Let us assume that we have computed the segmentation matrices $\hat{\mathbf{S}}_i$, $i = 1, \dots, I$ of the same signal. Segmentation results depend on the underlying signals and their SNR values and the choice of the algorithm parameters. In general, different signals and different SNR values require different algorithm parameters for achieving the best possible segmentation results. We would like to design a fusion method that will produce one segmentation matrix $\hat{\mathbf{S}}_{fu}$ from $\hat{\mathbf{S}}_i$, $i = 1, \dots, I$. Our goal is to design an algorithm that combines good sides of the different algorithm parameter choices. Thus, using this approach we hope to obtain a more robust algorithm.

Using multiple time resolutions. As already mentioned, any choice of the time resolution parameter T is a compromise between the two types of errors described earlier. One reasonable strategy is to apply the segmentation algorithm for several values of T which are appropriate for the given problem and then somehow fuse these segmentation results. By doing this we hope to exploit the good sides of each specific choice for T and thus, obtain a better algorithm. Let $\hat{\mathbf{S}}_i$ be the segmentation matrix computed for the time resolution T_i where $i = 1, \dots, I$ and $T_1 < T_2 < \dots < T_I$. In

order to perform the fusion we first form the matrix

$$\tilde{\mathbf{S}} = \begin{bmatrix} \hat{\mathbf{S}}_1 \\ \vdots \\ \hat{\mathbf{S}}_I \end{bmatrix} \quad (2.35)$$

and cluster its columns. Since $\tilde{\mathbf{S}}$ contains a finite number of distinct columns, the clustering task is simple. Each distinct column of $\tilde{\mathbf{S}}$ with all its repetitions defines one cluster and hence, produces one segmentation sequence. Some of these sequences represent true segments in the analyzed signal, while the others contain only very short impulses resulting from small disagreements in the estimated locations of the change points from $\hat{\mathbf{S}}_i$, $i = 1, \dots, I$. This impulse noise can be removed by applying our impulse noise filtering on $\tilde{\mathbf{S}}$. Let $\tilde{\mathbf{S}}_{filt} = [\tilde{\mathbf{s}}_{filt,1} \ \dots \ \tilde{\mathbf{s}}_{filt,K}]$ be the matrix computed by applying the filtering on $\tilde{\mathbf{S}} = [\tilde{\mathbf{s}}_1 \ \dots \ \tilde{\mathbf{s}}_K]$. The k -th column of $\tilde{\mathbf{S}}_{filt}$ is computed as

$$\tilde{\mathbf{s}}_{filt,k} = \text{majority vector}\{\tilde{\mathbf{s}}_{k-N_w}, \dots, \tilde{\mathbf{s}}_{k+N_w}\} \quad (2.36)$$

where $k = 1, \dots, K$. The matrix $\tilde{\mathbf{S}}_{filt}$ contains the initial segmentation results which will be refined in the next step.

In the final step we refine our initial estimates of the change points locations. For each change point in $\tilde{\mathbf{S}}_{filt}$ we perform the following procedure. Let k be the time index of one change point in $\tilde{\mathbf{S}}_{filt}$ which means that the k -th column is different from the $k-1$ -th column of $\tilde{\mathbf{S}}_{filt}$. We consider the window of time indices $[k - P_w, k + P_w]$ where P_w should be chosen so that the window contains only one change point. In each matrix $\hat{\mathbf{S}}_i$ we find the change point which is nearest to k and is from the window $[k - P_w, k + P_w]$. Let these change points be k_i for $i = 1, \dots, I$. Each point k_i is the time index of a transition from one cluster to another where the distance between these two clusters is d_i . Hence, for each k_i we assign the corresponding distance d_i for $i = 1, \dots, I$. If for some $\hat{\mathbf{S}}_j$ there is no change point in the window $[k - P_w, k + P_w]$ we set k_j to an arbitrary value and $d_j = 0$. Our final estimate of the change point location is

$$k_f = k_{\hat{i}} \quad (2.37)$$

where \hat{i} is the minimum value from $1 \leq i \leq I$ such that $d_i > d_{str}$ where d_{str} is a suitable threshold. In other words, if $d_1 > d_{str}$ then $k_f = k_1$. If $d_1 \leq d_{str}$ and $d_2 > d_{str}$ then $k_f = k_2$ and so on. If $d_i \leq d_{str}$ for all $1 \leq i \leq I$ then we set $k_f = k$.

The proposed fusion method for multiple time resolutions can be summarized as follows: (1) form the matrix $\tilde{\mathbf{S}}$ according to (2.35), (2) apply the impulse noise filtering on $\tilde{\mathbf{S}}$ which yields $\tilde{\mathbf{S}}_{filt}$, and (3) for each change point in $\tilde{\mathbf{S}}_{filt}$ perform the final estimation of its location using (2.37) which produces the final segmentation matrix $\hat{\mathbf{S}}_{fu}$.

Using multiple feature vectors. Let $\hat{\mathbf{S}}_i$ be the segmentation matrix computed for the i -th feature vector and the fixed time resolution T where $i = 1, \dots, I$. For example, these different feature vector could be obtained for different numbers of frequency bins J_i computed for the fixed time resolution T . Our first two steps are identical to the previous case. We form the matrix $\tilde{\mathbf{S}}$ according to (2.35) and apply the impulse noise filtering on $\tilde{\mathbf{S}}$ which yields $\tilde{\mathbf{S}}_{filt}$. Next, for each change point in $\tilde{\mathbf{S}}_{filt}$ perform the final estimation of its location as follows. Let k be the time index of one change point in $\tilde{\mathbf{S}}_{filt}$ and let us consider the window of time indices $[k - P_w, k + P_w]$. We define k_i and d_i for $i = 1, \dots, I$ in the same way as in the previous case. The final estimate of the change point location is

$$k_f = k_{i_{max}} \quad (2.38)$$

where

$$i_{max} = \operatorname{argmax}_{1 \leq i \leq I} d_i. \quad (2.39)$$

In other words, our final estimate is equal to the $k_{i_{max}}$ where $d_{i_{max}} = \max\{d_1, \dots, d_I\}$.

Using multiple sensors. One interesting application where fusion can be useful is to the case when $\hat{\mathbf{S}}_i$ for $i = 1, \dots, I$ are the segmentation matrices obtained at different sensors. In this case the received signals at different sensors are different because of the different channels between the transmitters and the sensors. However, if the sensors collect the received signals samples synchronously, the locations in time of change points seen at different sensors should differ only slightly because of the small difference in propagation delays of the channels between the transmitters and the sensors. Therefore, the fusion of the segmentation matrices $\hat{\mathbf{S}}_i$, $i = 1, \dots, I$ computed

at different sensors makes sense assuming that the sensors collect the received signals samples synchronously and that the differences in the propagation delays are much shorter than the duration of the statistically homogeneous segments in the received signals. If $\hat{\mathbf{S}}_i$ for $i = 1, \dots, I$ are computed with the same time resolution at each sensor we can use the same fusion method that was proposed for the multiple feature vectors. The method can be summarized as follows: (1) form the matrix $\tilde{\mathbf{S}}$ according to (2.35), (2) apply the impulse noise filtering on $\tilde{\mathbf{S}}$ which yields $\tilde{\mathbf{S}}_{filt}$, and (3) for each change point in $\tilde{\mathbf{S}}_{filt}$ perform the final estimation of its location using (2.38) and (2.39) which produces the final segmentation matrix $\hat{\mathbf{S}}_{fu}$.

In practice the fusion of the segmentation matrices $\hat{\mathbf{S}}_i$, $i = 1, \dots, I$ computed at different sensors can be performed at any sensor designated for this purpose or at some central computer which is collecting data from all sensors. Our fusion method requires the following data from each sensor: (1) its locally computed segmentation matrix $\hat{\mathbf{S}}_i$ and (2) for each pair (p, q) of locally detected clusters their distance $d_{p,q}$ defined by (2.29). The segmentation matrices $\hat{\mathbf{S}}_i$, $i = 1, \dots, I$ have a very simple structure with many repeated columns and hence, they can be compressed substantially. Therefore, the amount of data that needs to be transmitted from each sensor to the fusion node is relatively small.

2.3 Numerical examples

We illustrate the proposed algorithm with the following simulation example. In the example we consider a setup with one sensor and one source, whose locations are shown in Figure 2.1. Channel transfer function between the transmitter and the sensor is measured in the ORBIT room in WINLAB for the setup shown in Figure 2.1 [66]. This channel transfer function is used in our simulations. Thus, the real world propagation environment is faithfully reconstructed in the simulations.

The source is transmitting DBPSK signals with Barker sequence spreading which is used to produce 1 Mbit/s data rate in 802.11b systems [67]. Total observed bandwidth is $W = 20\text{MHz}$ and the sampling period is $T_s = 0.05\mu\text{s}$. Figure 2.2 shows the power trace

of the received signal at the sensor, where each point is the average power computed over $N_T = 200$ consecutive samples, which corresponds to the interval of $T = N_T T_s = 10\mu s$. The total observation time on this example is $5ms$. The transmission is happening in packets, where duration of each packet is $1ms$. After each packet there is a silent period which consists of the $50\mu s$ interframe space called DIFS and the contention window. Duration of the contention window is a random number of $20\mu s$ time slots where this random number is uniformly distributed and ranges between 0 and 31. These quantities are chosen according to the 802.11 standard [67]. At the sensor the signal is corrupted with additive white Gaussian noise. We define SNR as the ratio of the average received source signal power (when the source is on) at the sensor and the average sensor noise power. The SNR value in the example shown in Figure 2.2 is $30dB$.

We compute the feature vectors \mathbf{x}_n for the example shown in Figure 2.2 with $J = 2$ and apply the mean shift algorithm on this set of vectors where the scale parameters are selected using $k(N) = N^{0.65}$. Scatter plot of the vectors \mathbf{x}_n is shown in Figure 2.3. In this example there are two clusters: one consisting of the vectors \mathbf{x}_n corresponding to the time intervals with noise only and the other consisting of the vectors \mathbf{x}_n corresponding to the time intervals with signal plus noise. We also see from Figure 2.3 that the two clusters are connected. This connection is caused by the vectors \mathbf{x}_n corresponding to the time intervals containing statistical change points. Points of convergence of the mean shift algorithm are shown in Figure 2.3 with red marks. These points are local maxima of the pdf estimate (2.7) computed from the vectors \mathbf{x}_n . We see from Figure 2.3 that there are five local maxima of the pdf estimate. Using the result of the mean shift clustering we compute the segmentation sequences which are shown in Figure 2.4. Obviously, the mean shift algorithm did not perform a correct segmentation in this example because of the spurious peaks in the pdf estimate. After applying our cluster validation procedure with $d_{tr} = 2$ on the results of the mean shift algorithm we find that there are two clusters in the vectors \mathbf{x}_n which is correct. Using the clustering obtained after the cluster validation procedure we obtain the segmentation sequences shown in Figure 2.5. By comparing Figure 2.2 and Figure 2.5 we see that the algorithm performed a reasonably accurate segmentation.

Now we repeat this experiment for a range of SNR values. For each SNR point we repeat the experiment 100 times and calculate the percentage of times when the algorithm finds the correct number of clusters. These percentages are computed before and after the cluster validation procedure. The results are shown in Figure 2.6. We see that the mean shift algorithm finds the correct number of clusters for $-5dB < SNR < 20dB$. For $SNR < -5dB$ the two clusters are very close to each other and cannot be resolved. For $SNR > 20dB$ the mean shift algorithm overestimates the number of clusters because of the spurious peaks in the pdf estimate. After adding the cluster validation procedure we obtain an algorithm that finds the correct number of clusters for a much wider SNR range. In general, the mean shift algorithm treats every local maximum of the pdf estimate as one of the pdf modes and creates a separate cluster for it. Since the pdf is always estimated from a finite sample size, some local maxima may be artifacts of the pdf estimation and they do not represent any real clusters in the data. Because of this problem it is very difficult to obtain reliable clustering using the mean shift algorithm alone. In order to obtain more reliable clustering the mean shift algorithm results must go through some form of cluster validation where all clusters caused by spurious peaks should be identified and merged into real clusters.

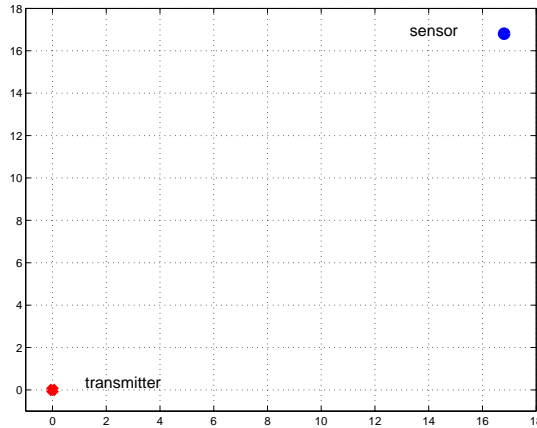


Figure 2.1: Locations of the transmitter and sensor

Now we discuss the role of the proposed impulse noise removal procedure. Again we consider the setup shown in Figure 2.1. Figure 2.8 shows power trace of the received signal for $SNR = -3dB$. Scatter plot of the vectors \mathbf{x}_n for this example is shown in

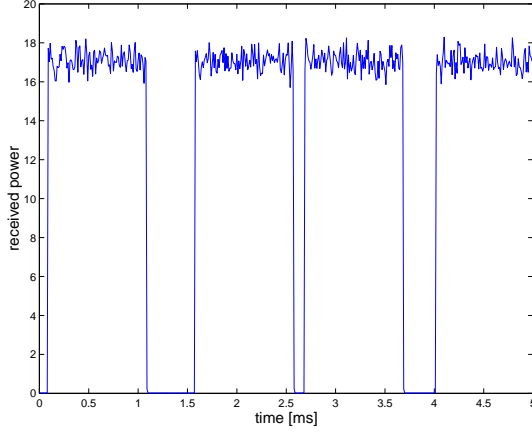


Figure 2.2: Power trace of the received signal computed with time resolution $T = 10\mu s$ for $SNR = 30dB$

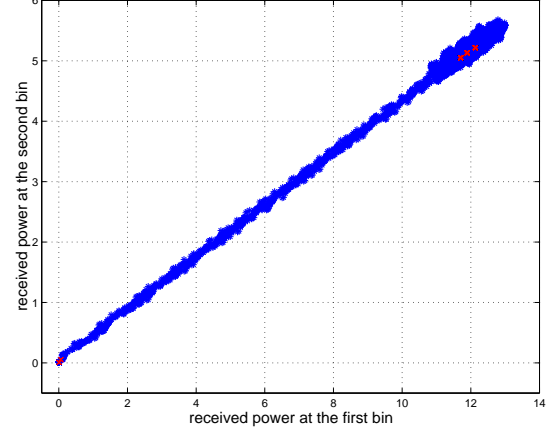


Figure 2.3: Scatter plot of the feature vectors and local maxima (red marks) of the pdf estimate

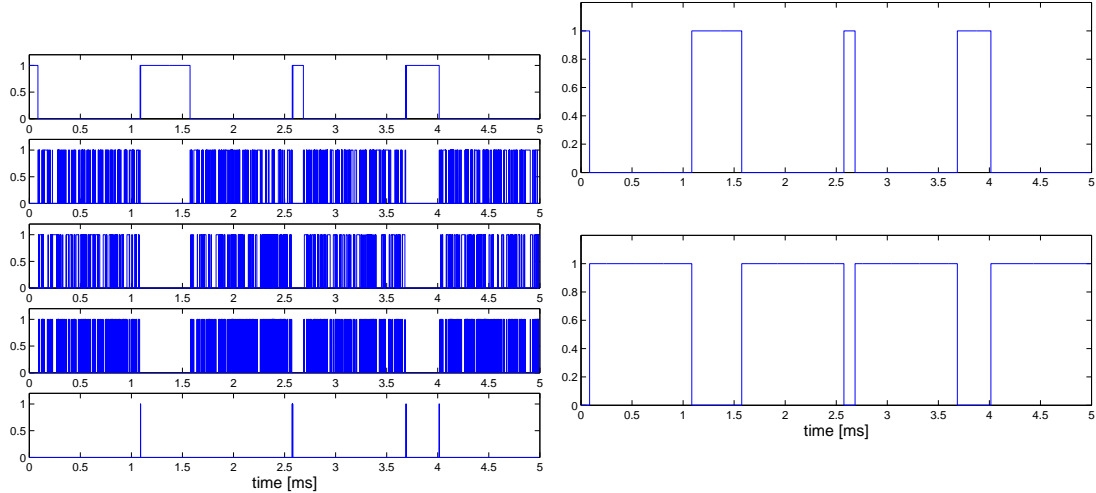


Figure 2.4: Segmentation sequences obtained using mean shift clustering

Figure 2.5: Segmentation sequences obtained after the cluster validation procedure

Figure 2.9. We see from Figure 2.9 that the algorithm correctly found two clusters. Segmentation results before the impulse noise removal are shown in Figure 2.10 where we see some amount of impulse noise. These segmentation results are obtained by clustering the vectors \mathbf{x}_n where each vector is assigned to one of the clusters independently of all other vectors. At low SNR values some vectors may be assigned incorrectly which

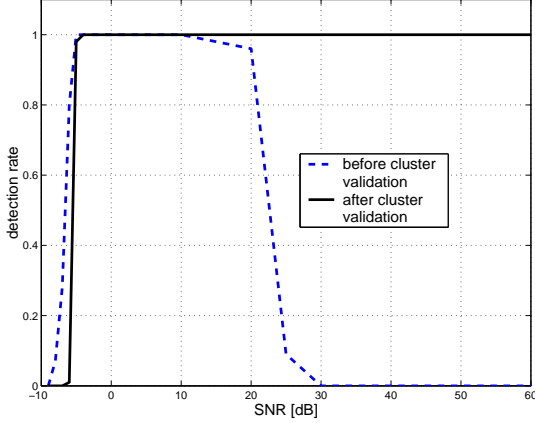


Figure 2.6: Detection rates for the correct number of clusters before and after the cluster validation procedure

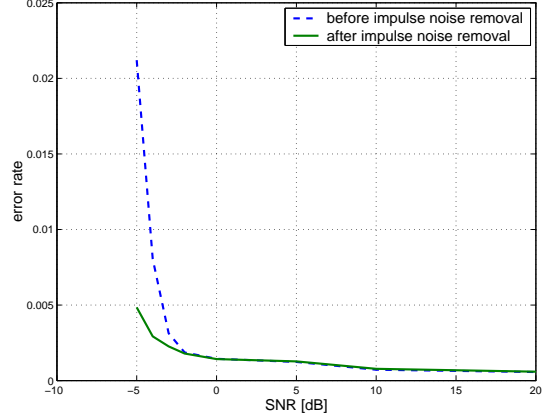


Figure 2.7: Segmentation error rate before and after the impulse noise removal for $J = 2$ and $T = 10\mu s$

causes impulse noise in the segmentation sequences shown in Figure 2.10. After applying our impulse noise removal procedure with $N_w = 500$ we obtain the segmentation results shown in Figure 2.11 where we see that the impulse noise has been removed.

Next, we repeat this experiment for a range of SNR values. At each SNR point we repeat the experiment 100 times where in each run we generate different signal and noise realizations and find segmentation sequences before and after the impulse noise removal procedure. We calculate segmentation error rate for each run by comparing the true segmentation sequences with the segmentation sequences estimated by our algorithm. Segmentation error rate for each SNR point is computed as an average value of the rates obtained for the runs at that SNR value. These segmentation error rates before and after impulse noise removal are shown in Figure 2.7. We see that as SNR increases the segmentation error rate decreases. However, at high SNR values this decrease stops and the error rate becomes approximately constant. In our algorithm we analyze signals with a window of the finite size N_T and hence, we cannot perfectly localize the signal segments in time. This fact explains why the segmentation error rate does not tend to zero at high SNR values. When $SNR < -3dB$ the two clusters start to overlap and the error rate rapidly increases. At high SNR values error rates before and after impulse noise removal are very similar. At low SNR values the error rate after filtering increases

much more slowly than the error rate before filtering because our impulse noise removal procedure can correct some errors caused by the cluster overlap.

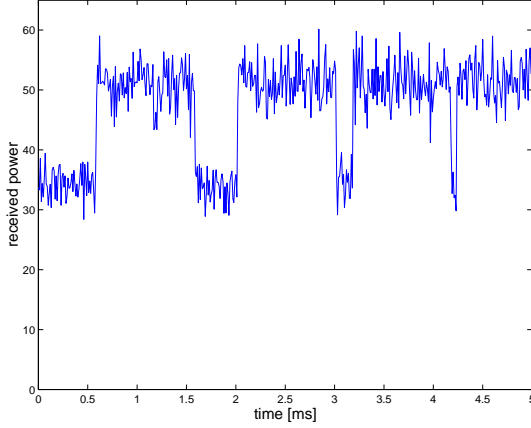


Figure 2.8: Power trace of the received signal computed with time resolution $T = 10\mu s$ for $SNR = -3dB$

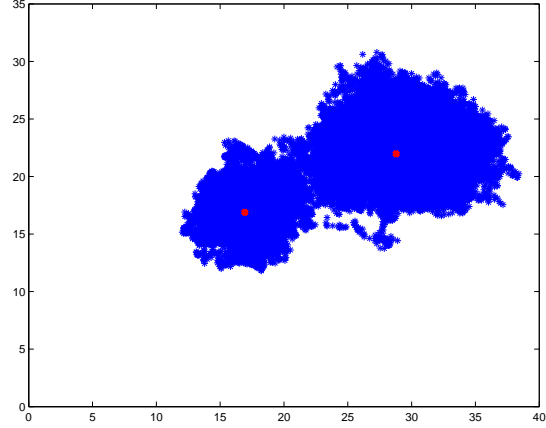


Figure 2.9: Scatter plot of the feature vectors and local maxima (red marks) of the pdf estimate

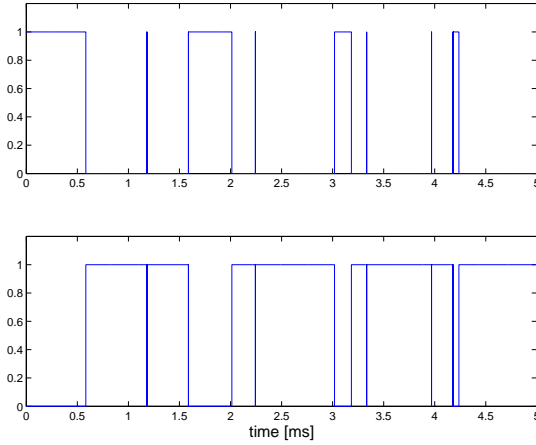


Figure 2.10: Segmentation results before the impulse noise removal

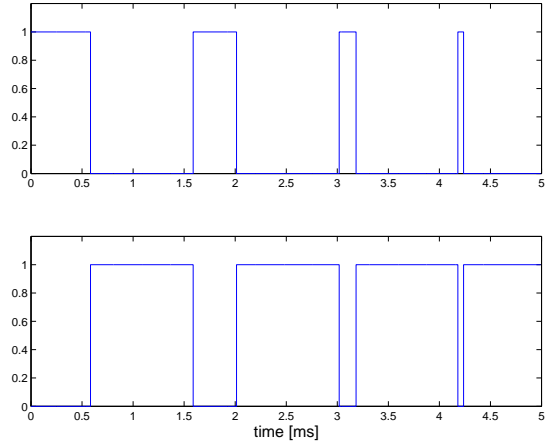


Figure 2.11: Segmentation results after the impulse noise removal

Now we study effects of the choice of the time resolution parameter T . Again, we consider the same setup shown in Figure 2.1 with the same signals as described earlier. We apply our segmentation algorithm for $J = 2$ and three different values of the time resolution parameter T : $5\mu s$, $10\mu s$, and $20\mu s$. For each time resolution we run the algorithm for a range of SNR values. At each SNR point we compute the

average detection rate of determining the correct number of clusters and the average segmentation error rate. Figure 2.12 shows the detection rates of determining the correct number of clusters. We see that these curves exhibit a threshold behavior. When SNR is above some threshold the detection rate is one. When SNR is below the threshold the algorithm cannot resolve two closely spaced clusters and the detection rate rapidly falls to zero. We see from Figure 2.12 that the SNR threshold becomes lower as the time resolution increases. Main limiting factor in resolving two closely spaced clusters is the PSD estimation error caused by the finite sample size. As T increases this error reduces and hence, the SNR threshold also reduces. Figure 2.13 shows the segmentation error rates for different time resolution. We see that at high SNR values the error rates cannot fall below certain error floor rates. These error floor rates are caused by the fact that we are working with finite time resolution which prevents perfect localization of signals in time. As T increases our ability to localize signals in time reduces and hence, the error floor rate increases. When SNR approaches the threshold value the corresponding error rate rapidly increases because of the cluster overlap. We conclude that a small T is good at high SNR values because it reduces the error floor rate, while a large T reduces the threshold SNR. In order to obtain an algorithm that combines positive effects of a large T at low SNR values and a those of a small T at high SNR values we perform a fusion of the segmentation results for different time resolutions using the proposed fusion method. Figure 2.12 shows that the fusion curve of detection rates of determining the correct number of clusters is approximately the same as the curve obtained for the largest value of T , which is satisfactory. Figure 2.13 shows the fusion curve of the segmentation error rates. We see that this curve is approximately an envelope of the three curves obtained for different time resolutions, which is the desired behavior.

Now we study effects of the choice of the number of frequency bins J . Again, we consider the same setup shown in Figure 2.1 with the same signals as described earlier. For a fixed time resolution T the number of frequency bins is in the range $1 \leq J \leq N_T$ where $N_T = T/T_s$. For $T = 10\mu s$ and $T_s = 0.05\mu s$ we get $N_T = 200$ which implies $1 \leq J \leq 200$. Since J is the length of \mathbf{x}_n any increase of J increases

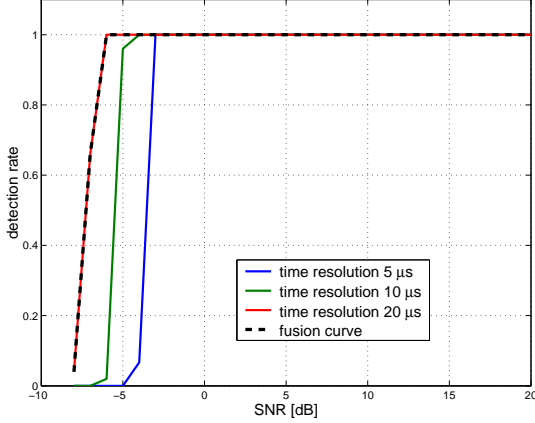


Figure 2.12: Detection rates of the correct number of clusters for different time resolutions

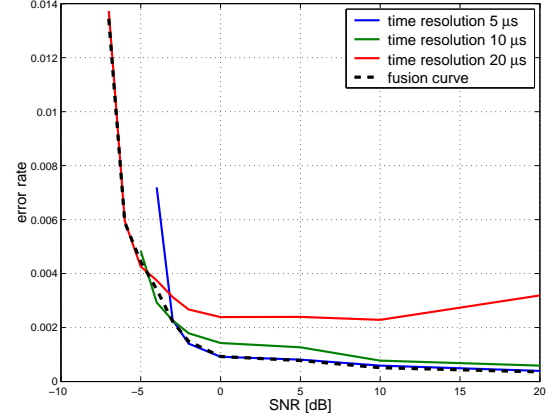


Figure 2.13: Segmentation error rates for different time resolutions

computational complexity of our algorithm. Another limitation on J stems from the fact that our approach is based on nonparametric pdf estimation. It is well known that nonparametric pdf estimation becomes very difficult if the number of dimensions is too high. For example, it is suggested in [47, sec. 7.2] that kernel pdf estimate is not reliable if the number of dimensions is more than five. Therefore, a practically usable range for J is very limited because of increased computational complexity and problems with nonparametric pdf estimation for a high J . We apply our segmentation algorithm for $T = 10\mu s$ and three different numbers of frequency bins J : 1, 2, and 5. Results are shown in Figures 2.15 and 2.21. At low SNR values the performances for $J = 1$ and $J = 2$ are very similar, while for $J = 5$ we have a somewhat lower threshold SNR. At high SNR values all three values of J produce very similar performances. Next, we perform a fusion of the segmentation results obtained for $J = 1$, $J = 2$, and $J = 5$. We see from Figures 2.15 and 2.21 that the fusion curves approximately follow the curves obtained for $J = 1$ and $J = 2$, which is good since the performance obtained for $J = 1$ and $J = 2$ is better than that obtained for $J = 5$.

In the next example we consider the setup with one transmitter and four sensors shown in Figure 2.16. The transmitter transmits the same signal as described earlier. Channel transfer functions between the transmitter and the sensors are measured in

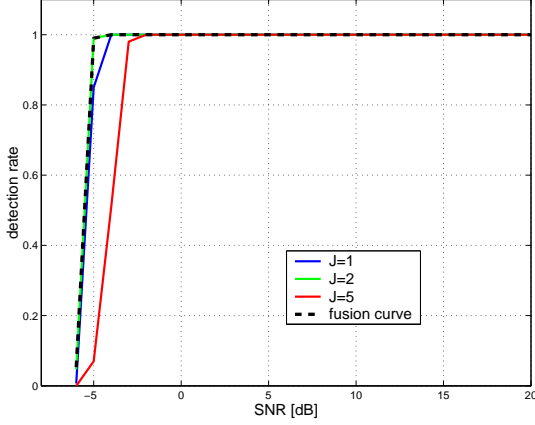


Figure 2.14: Detection rates of the correct number of clusters for different numbers of frequency bins

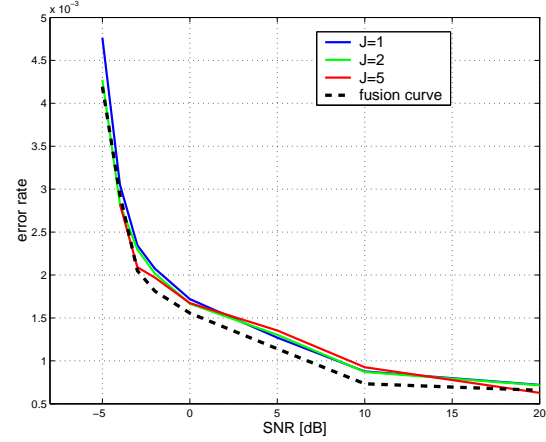


Figure 2.15: Segmentation error rates for different numbers of frequency bins

the ORBIT room in WINLAB for the setup shown in Figure 2.16 [66]. These channel transfer functions are used in our simulations. First, we consider an example where the received signals at the sensors have the power traces shown in Figure 2.17. The SNR values at the sensors are: $SNR_1 = 0.42dB$, $SNR_2 = -2.46dB$, $SNR_3 = -8dB$, and $SNR_4 = -2.46dB$. These significant SNR differences are caused by differences in the channel transfer functions. First, we apply our segmentation algorithm on the received signal at each sensor separately. The algorithm parameters were $T = 10\mu s$, $J = 2$, $k(N) = N^{0.65}$, and $N_w = 500$. Segmentation sequences obtained by applying the algorithm separately at each sensor are shown in Figure 2.10 where different colors were used to plot the sequences from different sensors. We see that sensor 1, sensor 2, and sensor 4 correctly detected two clusters while sensor 3 could not resolve the two closely spaced clusters because its SNR was too low. Now we would like to perform a fusion of the locally computed segmentation sequences. Let $\hat{\mathbf{S}}_i$, $i = 1, 2, 3, 4$ be the locally computed segmentation sequences. We form the matrix $\tilde{\mathbf{S}} = [\hat{\mathbf{S}}_1^T \ \hat{\mathbf{S}}_2^T \ \hat{\mathbf{S}}_3^T \ \hat{\mathbf{S}}_4^T]^T$ and cluster the columns of $\tilde{\mathbf{S}}$. Figure 2.20 shows segmentation sequences obtained after this simple fusion of the locally computed segmentation sequences. We see that this simple fusion produced 8 clusters and hence, there are 8 segmentation sequences shown in Figure 2.20. However, only the first and the last row in Figure 2.17 represent true

clusters while all other sequences contain only impulse noise caused by small disagreements in locally estimated statistical change points. Therefore, this simple fusion is not satisfactory and its results must be further processed. After impulse noise removal and final estimation of locations of the statistical change points we obtain the segmentation sequences shown in Figure 2.19. We see that the algorithm correctly found two segmentation sequences and that the segmentation is reasonably accurate.

Next, we study the performance for a range of SNR values. At each SNR value we repeat the above experiment 300 times and compute the average detection rate of determining the correct number of clusters and the average segmentation error rate. These rates are computed for each sensor separately and for the results computed using our fusion method. The results are shown in Figure 2.21 and 2.22. The SNR values at the sensors are $SNR_1 = SNR + 8.42dB$, $SNR_2 = SNR + 5.44dB$, $SNR_3 = SNR$, and $SNR_4 = SNR + 5.44dB$ where SNR is the variable from the graphs shown in Figure 2.21 and 2.22. We see that the performances at different sensors differ significantly which is caused by significant differences in the corresponding SNR values. In other words, the same transmit power level produces very different received signal powers at different sensors because of differences in the corresponding channel transfer functions. We also see from Figure 2.21 and 2.22 that performance of the fusion method is very similar to that of the sensor with the highest SNR. Thus, by adding multiple sensors and using this fusion method we can extend the range of the transmit power for which our algorithm correctly identifies the number of clusters. We conclude that exploiting spatial diversity provided by multiple sensors is a very effective method for improving performance of the algorithm.

Once we have localized in time statistically homogeneous segments in the received signal we can further analyze them. Let us illustrate this using as an example the received signal whose power trace is shown in Figure 2.8. For each of the localized segments we will estimate the second order spectrum $S_{2x}(f_1, f_2)$. For the signal $x(t)$ admitting Cramer spectral representation [68]

$$x(t) = \int_{-\infty}^{\infty} e^{j2\pi f t} dX(f) \quad (2.40)$$

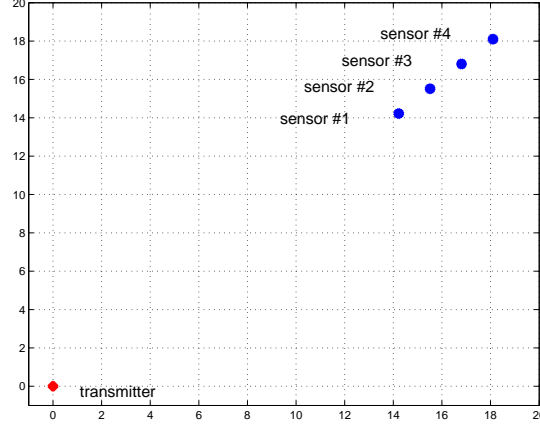


Figure 2.16: Locations of the transmitter and the sensors

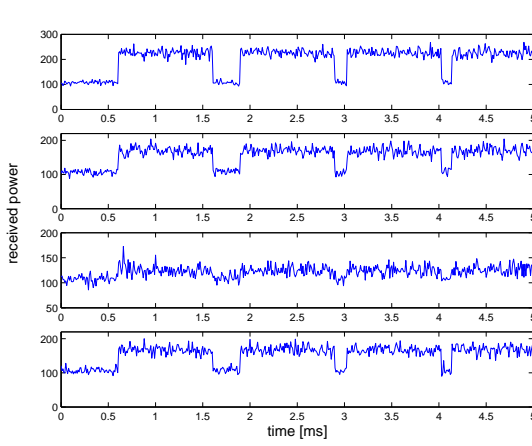


Figure 2.17: Received power traces at different sensors

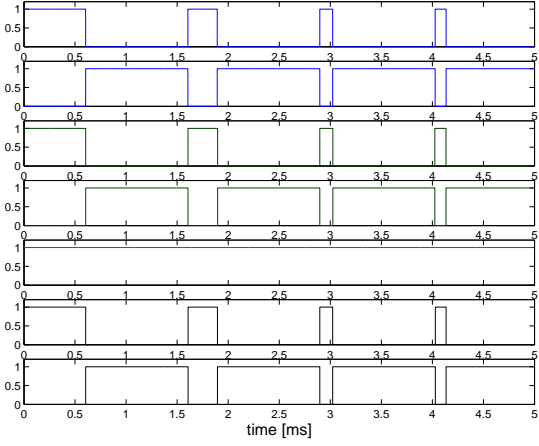


Figure 2.18: Segmentation sequences obtained at different sensors

the second order spectrum $S_{2x}(f_1, f_2)$ of $x(t)$ is defined as

$$S_{2x}(f_1, f_2)df_1df_2 = E[dX(f_1)dX(f_2)^*]. \quad (2.41)$$

For $f_1 = f_2$ we have $S_{2x}^{(0)}(f)df = E[|dX(f)|^2]$ where $S_{2x}^{(0)}(f) = S_{2x}(f, f)$ represents the power spectrum of $x(t)$. For $f_1 \neq f_2$, $S_{2x}(f_1, f_2)$ represents correlation between the spectral components at f_1 and f_2 . Let us define the autocorrelation function $R_{2x}(t, \tau) = E[x(t + \tau)x(t)^*]$. If $x(t)$ is a second order stationary signal (i. e., $R_{2x}(t, \tau) = R_{2x}(\tau)$) then $S_{2x}(f_1, f_2) = 0$ for $f_1 \neq f_2$ [69]. If $x(t)$ is a second order cyclostationary signal with the period T_c (i. e., $R_{2x}(t + T_c, \tau) = R_{2x}(t, \tau)$) then $S_{2x}(f_1, f_2) = 0$ for $f_1 \neq f_2 + m/T_c$ where $m = \dots, -2, -1, 0, 1, 2, \dots$. For $f_1 = f_2 + m/T_c$ we define the spectra

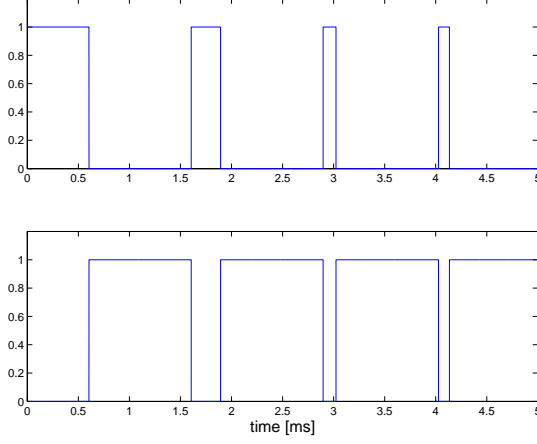


Figure 2.19: Segmentation sequences computed as the final result of the fusion of the sequences from different sensors

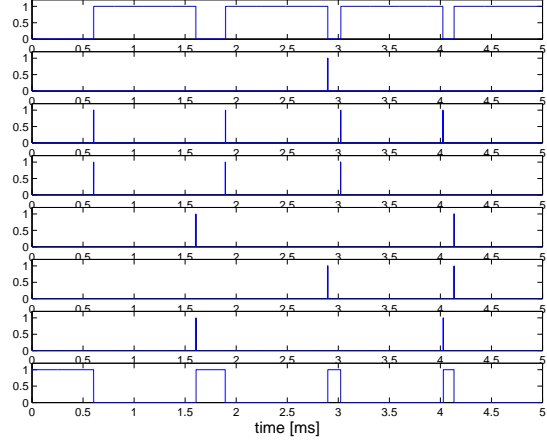


Figure 2.20: Segmentation sequences obtained by a simple fusion of the sequences from different sensors

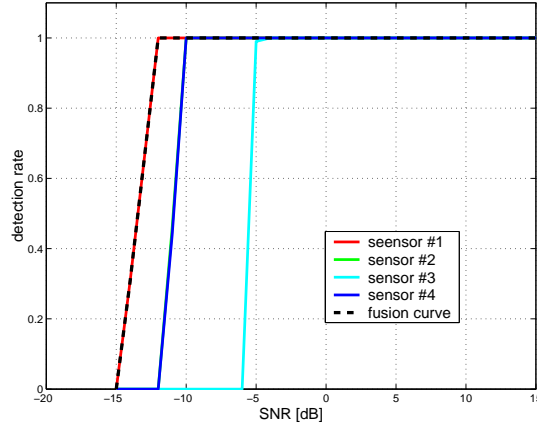


Figure 2.21: Detection rates of the correct number of clusters for different sensors

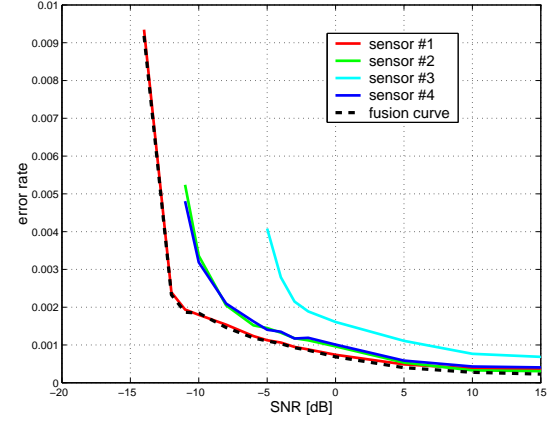


Figure 2.22: Segmentation error rates for different sensors

$S_{2x}^{(m)}(f)df = E[dX(f)dX(f - m/T_c)^*]$ where $S_{2x}^{(m)}(f) = S_{2x}(f, f - m/T_c)$ for $m = \dots, -2, -1, 0, 1, 2, \dots$. For $m = 0$ we obtain the power spectrum $S_{2x}^{(0)}(f)$ and for $m \neq 0$ we obtain the second order cyclic spectra $S_{2x}^{(m)}(f)$ [10].

Let us now estimate $S_{2x}(f_1, f_2)$ for the first segment of the signal whose power trace is shown in Figure 2.8. This segment starts at 0 and ends at about $0.6ms$ and it contains

only stationary white Gaussian noise. The estimate of $S_{2x}(f_1, f_2)$ for the first segment is shown in Figure 2.23. We see that the signal has a flat power spectrum and the estimated spectrum is close to zero for $f_1 \neq f_2$. Next, we estimate $S_{2x}(f_1, f_2)$ for the second segment which starts at about $0.6ms$ and ends at about $1.6ms$. This segment contains DBPSK signal with Barker sequence spreading plus stationary white Gaussian noise. The DBPSK signal with Barker sequence spreading is a linearly modulated signal with the symbol period $T_c = 1\mu s$ and it is cyclostationary with this period. The estimate of $S_{2x}(f_1, f_2)$ for the second segment is shown in Figure 2.24. For $f_1 = f_2$ we get the power spectrum which consists of the DBPSK signal power spectrum and the noise power spectrum. For $f_1 = f_2 + m/T_c$ and $m \neq 0$ we get the cyclic spectra of the DBPSK signal. We see that the spectrum $S_{2x}(f_1, f_2)$ provides a useful characterization of the signals in the observed frequency band. In this example the analysis based on $S_{2x}(f_1, f_2)$ can be used only if the statistically homogeneous segments in the received signal have been localized in time.

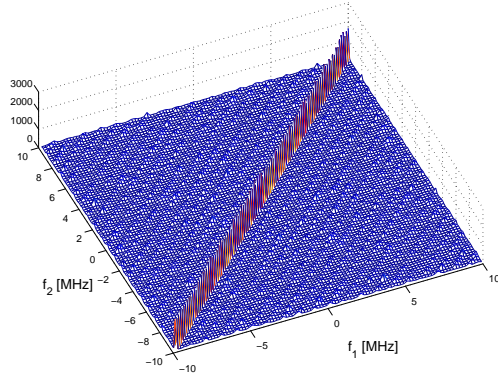


Figure 2.23: Spectrum $S_{2x}(f_1, f_2)$ for the first noise segment

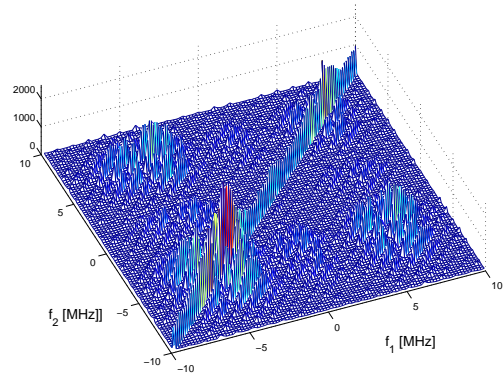


Figure 2.24: Spectrum $S_{2x}(f_1, f_2)$ for the first signal plus noise segment

2.4 Conclusion

We have proposed a method for localization in time of statistically homogeneous segments in the received signal. The proposed method is almost completely nonparametric and it does not assume any probability distributions for the analyzed signals. This is

an important advantage since these probability distributions are almost never known in real world applications. In this application we did not impose any parametric models on the underlying signals. However, if such models are appropriate they can be easily incorporated into the algorithm. Thus, any apriori knowledge about the signals can be exploited. We have also proposed a method for fusion of segmentation decisions obtained using different time resolutions or different feature vectors. This is a useful technique for obtaining a more robust algorithm. The proposed segmentation algorithm was designed for one specific application but it can be applied in many other situations where it is desired to find statistical change points in a piecewise statistically homogeneous signal.

Chapter 3

Fourth Order Spectrum Based Analysis

Using the proposed segmentation algorithm we can localize in time segments in the received signal generated by the same combination of active transmitted signals and sensor noise. After these segments have been localized in time they can be further analyzed using various statistical methods. The goal of this analysis is to extract more information about the signals using the observed frequency band. In this section we propose one signal analysis method that uses partial information from the fourth order spectrum of the localized segments. Main goal of this method is to estimate the spectrum and the associated activity sequence in time for each signal present in the observed frequency band. This type of analysis is useful when the transmitted signals are either linearly modulated or they can be represented as a sum of a finite number of linearly modulated terms.

3.1 Signal analysis method

Let us first introduce a few definitions. Let $x(t)$ be a random process admitting Cramer spectral representation

$$x(t) = \int_{-\infty}^{\infty} e^{j2\pi f t} dX(f) \quad (3.1)$$

where $dX(f)$ is called spectral function [68] or spectral process [70]. The second order spectrum (power spectrum) of $x(t)$ is defined as

$$S_{2x}(f)df = E[|dX(f)|^2] \quad (3.2)$$

and the fourth order spectrum (trispectrum) is defined as

$$\begin{aligned}
S_{4x}(f_1, f_2, f_3)df_1df_2df_3 = \\
cum(dX(f_1), dX(f_2), dX(-f_3)^*, dX(f_1 + f_2 + f_3)^*) = \\
E[dX(f_1)dX(f_2)dX(-f_3)^*dX(f_1 + f_2 + f_3)^*] \\
- E[dX(f_1)dX(f_2)]E[dX(-f_3)^*dX(f_1 + f_2 + f_3)^*] \\
- E[dX(f_1)dX(-f_3)^*]E[dX(f_2)dX(f_1 + f_2 + f_3)^*] \\
- E[dX(f_1)dX(f_1 + f_2 + f_3)^*]E[dX(f_2)dX(-f_3)^*]
\end{aligned} \tag{3.3}$$

where we used the definition of the fourth order cumulants [36]. The spectrum $S_{4x}(f_1, f_2, f_3)$ is related to the fourth order statistic

$$C_{4x}(t, \tau_1, \tau_2, \tau_3) = cum(x(t + \tau_1), x(t + \tau_2), x(t + \tau_3)^*, x(t)^*). \tag{3.4}$$

If $x(t)$ is stationary, we have $C_{4x}(t, \tau_1, \tau_2, \tau_3) = C_{4x}(\tau_1, \tau_2, \tau_3)$ and then

$$S_{4x}(f_1, f_2, f_3) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} C_{4x}(\tau_1, \tau_2, \tau_3) e^{-j2\pi(f_1\tau_1 + f_2\tau_2 + f_3\tau_3)} d\tau_1 d\tau_2 d\tau_3. \tag{3.5}$$

If $x(t)$ is cyclostationary with the period T , we have

$$C_{4x}(t + T, \tau_1, \tau_2, \tau_3) = C_{4x}(t, \tau_1, \tau_2, \tau_3) = \sum_{q=-\infty}^{\infty} C_{4x}^{(q)}(\tau_1, \tau_2, \tau_3) e^{j2\pi\frac{q}{T}t} \tag{3.6}$$

where $C_{4x}^{(q)}(\tau_1, \tau_2, \tau_3) = \frac{1}{T} \int_0^T C_{4x}(t, \tau_1, \tau_2, \tau_3) e^{-j2\pi\frac{q}{T}t} dt$. In this case we have

$$S_{4x}(f_1, f_2, f_3) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} C_{4x}^{(0)}(\tau_1, \tau_2, \tau_3) e^{-j2\pi(f_1\tau_1 + f_2\tau_2 + f_3\tau_3)} d\tau_1 d\tau_2 d\tau_3. \tag{3.7}$$

These relations between $S_{4x}(f_1, f_2, f_3)$ and $C_{4x}(t, \tau_1, \tau_2, \tau_3)$ can be shown using the methodology from [70]. When $x(t)$ is complex we can define different statistics for different conjugation patterns in (3.4). The spectrum $S_{4x}(f_1, f_2, f_3)$ defined in (3.3) is useful for our application.

In our work we are going to use the following two dimensional slice of the fourth order spectrum

$$\tilde{S}_{4x}(f, v) = S_{4x}(f, v, -v) \tag{3.8}$$

which using the definition (3.3) can be written as

$$S_{04x}(f, v)dfdv = cum(dX(f), dX(v), dX(v)^*, dX(f)^*) = \\ E[|dX(f)dX(v)|^2] - |E[dX(f)dX(v)^*]|^2 - |E[dX(f)dX(v)]|^2 - E[|dX(f)|^2]E[|dX(v)|^2]. \quad (3.9)$$

The quantity $S_{04x}(f, v)$ for $f = v$ is the fourth order autocumulant (kurtosis) at frequency f . It is zero for Gaussian signals and non-zero for most non-Gaussian signals. Thus, it can be interpreted as deviation from Gaussianity at frequency f . Kurtosis has also other interpretations and it is difficult to determine exactly what this statistic measures [71]. The interpretation of kurtosis is even more complicated for complex signals [70], which is the case in our application. Kurtosis can also be defined for some classes of nonstationary signals where it has a different interpretation [40]. The quantity $S_{04x}(f, v)$ for $f \neq v$ measures a certain form of statistical dependency between spectral components at frequencies f and v . Again, it is not easy to exactly understand the nature of this statistical dependency. In contrast to the fourth order spectrum, the second order spectrum $S_{2x}(f)$ measures only average power at frequency f and contains no information about statistical dependencies between spectral components at different frequencies. Hence, the slice $S_{04x}(f, v)$ contains significant information about the received signal, which is not present in its second order spectrum.

We now consider the scenario where one sensor observes a frequency band used by M packet based radio transmitters. The received signal $r(t)$ is given by (2.1). Let us assume that the received signal consists of I_{cl} sets of one or more continuous time segments where the segments from each set are generated by the same combination of active transmitted signals and thus, have the same fourth order spectrum slice $S_{04x}(f, v)$. The problem of localizing these segments in time has been treated in the previous chapter. We note that each set of segments mentioned above corresponds to one cluster in our segmentation algorithm. Let $S_{04}(f, v, i)$ be the fourth order spectrum slice defined in (3.9) for the i -th cluster. Assuming statistically independent signals over each segment we have

$$S_{04}(f, v, i) = \sum_{m=1}^M |H_m(f)|^2 |H_m(v)|^2 S_{04m}(f, v) c_{im} + S_{04n}(f, v) \quad (3.10)$$

where $S_{04m}(f, v)$ is the fourth order spectrum slice of the m -th transmitted signal and $c_{im} \in \{0, 1\}$ indicates whether the m -th source is on during the segments of the i -th cluster for $i = 1, \dots, I_{cl}$. We define the matrix $\mathbf{C} = [c_{im}]$ of the size I_{cl} by M . The columns of \mathbf{C} are activity sequences associated with different signals where these sequences are functions of the cluster index i . We also define the matrix $\mathbf{B} = [b_{km}]$, which contains the activity sequences as functions of the time index $k = 1, \dots, K$. We have $\mathbf{B} = \mathbf{S}^T \mathbf{C}$ where \mathbf{S} is the segmentation matrix of the size I_{cl} by K defined earlier. Initially, we will assume the receiver noise is Gaussian and hence, $S_{04n}(f, v) = 0$. Under some assumptions our approach can be extended to nongaussian noise. This will be explained later. Let us define the three-way array \mathbf{Y} with entries

$$\mathbf{Y}(j, n, i) = \int_{(j-1)\Delta f}^{j\Delta f} \int_{(n-1)\Delta f}^{n\Delta f} S_{04}(f, v, i) df dv = \sum_{m=1}^M q_{jnm} c_{im} \quad (3.11)$$

where $q_{jnm} = \int_{(j-1)\Delta f}^{j\Delta f} \int_{(n-1)\Delta f}^{n\Delta f} |H_m(f)|^2 |H_m(v)|^2 S_{04m}(f, v) df dv$ for $j, n = 1, \dots, J_{fb}$. The number of frequency bins is $J_{fb} = W/\Delta f$, where W is the total observed bandwidth and Δf is the frequency discretization step. Let \mathbf{Y}_i be the J_{fb} by J_{fb} matrix obtained by fixing the index i in (3.11). We have

$$\mathbf{Y}_i = \sum_{m=1}^M \mathbf{Q}_m c_{im} \quad (3.12)$$

where $\mathbf{Q}_m = [q_{jnm}]$ is the fourth order spectrum matrix of the m -th signal. These matrices depend on transmitted signal formats. We consider two classes of transmitted signals.

Linear modulation formats. We now consider an important class of linear modulations where the m -th transmitted signal has the following form

$$x_m(t) = \sum_{k=-\infty}^{\infty} a_{km} p_m(t - kT_m) \quad (3.13)$$

where a_{km} is an i.i.d. sequence of input symbols, $p_m(t)$ is the pulse shape, and T_m is the symbol period. Using the definition it can be shown that the fourth order spectrum of $x_m(t)$ is

$$S_{4m}(f_1, f_2, f_3) = \frac{S_{4am}(f_1, f_2, f_3)}{T_m} P_m(f_1) P_m(f_2) P_m(-f_3)^* P_m(f_1 + f_2 + f_3)^* \quad (3.14)$$

where $P_m(f) = \int_{-\infty}^{\infty} p_m(t) e^{-j2\pi ft} dt$. Since a_{km} is an i.i.d. sequence $S_{4am}(f_1, f_2, f_3) = \gamma_{4am} = \text{cum}(a_{km}, a_{km}, a_{km}^*, a_{km}^*)$. Using (3.9) we have

$$S_{04m}(f, v) = k_m |P_m(f)|^2 |P_m(v)|^2 \quad (3.15)$$

where $k_m = \gamma_{4am}/T_m$. It follows that $q_{jnm} = k_m f_{jm} f_{nm}$, where $f_{nm} = \int_{(n-1)\Delta f}^{n\Delta f} |H_m(f)|^2 |P_m(f)|^2 df$. We see that for linear modulation formats \mathbf{Q}_m is of rank one. In this case the entries of \mathbf{Y} are

$$\mathbf{Y}(j, n, i) = \sum_{m=1}^M k_m f_{jm} f_{nm} c_{im}. \quad (3.16)$$

Now we observe that

$$\mathbf{Y}_i = \mathbf{F} \mathbf{\Lambda}_i \mathbf{F}^T \quad (3.17)$$

where $\mathbf{F} = [f_{jm}]$ and $\mathbf{\Lambda}_i = \text{diag}([k_1 c_{i1} \quad \dots \quad k_M c_{iM}])$. We see that the matrix \mathbf{F} jointly diagonalizes the matrices \mathbf{Y}_i for $i = 1, \dots, I_{cl}$. Under certain conditions the matrix \mathbf{F} can be recovered from the set of matrices \mathbf{Y}_i for $i = 1, \dots, I_{cl}$. The problem of finding \mathbf{F} from the set of matrices \mathbf{Y}_i for $i = 1, \dots, I_{cl}$ is known as joint diagonalization by congruence [72]. After \mathbf{F} has been recovered it is possible to find contributions of different signals to \mathbf{Y} . Thus, using this method we can estimate spectra and activity sequences of the signals in the observed frequency band.

Let us explain how \mathbf{F} can be recovered from \mathbf{Y}_i for $i = 1, \dots, I_{cl}$ assuming that $J_{fb} \geq M$ and $\text{rank}(\mathbf{F}) = M$. When $J_{fb} > M$ we can apply dimension reduction and reduce the problem to $J_{fb} = M$. Hence, we can assume $J_{fb} = M$ and $\text{rank}(\mathbf{F}) = M$. We define $\mathbf{Y}_0 = \sum_{i=1}^{I_{cl}} \alpha_i \mathbf{Y}_i = \mathbf{F} \mathbf{\Lambda}_0 \mathbf{F}^T$ where the vector $\boldsymbol{\alpha} = [\alpha_1 \quad \dots \quad \alpha_{I_{cl}}]$ is chosen so that $\mathbf{\Lambda}_0$ is invertible. The vector $\boldsymbol{\alpha}$ should be chosen so that $\boldsymbol{\alpha} \in S_t \setminus \{S_1 \cup \dots \cup S_M\}$ where S_t is the entire I_{cl} -dimensional space and S_m is the $I_{cl} - 1$ -dimensional subspace orthogonal to the m -th column of \mathbf{C} . Since this union of $I_{cl} - 1$ -dimensional subspaces cannot fill the entire I_{cl} -dimensional space we know that an appropriate vector $\boldsymbol{\alpha}$ exists. Therefore it is possible to create an invertible $\mathbf{\Lambda}_0$ unless $k_m = 0$ or $c_{1m} = \dots = c_{I_{cl},m} = 0$ for some $1 \leq m \leq M$. If this is the case we can remove the m -th columns from \mathbf{F} and \mathbf{C} and consider decomposition of \mathbf{Y} into $M - 1$ terms. Hence, we can assume that $\mathbf{\Lambda}_0$ is invertible. We form the matrices

$$\mathbf{Y}_i \mathbf{Y}_0^{-1} = \mathbf{F} \mathbf{\Lambda}_i \mathbf{\Lambda}_0^{-1} \mathbf{F}^{-1} \quad (3.18)$$

for $i = 1, \dots, I_{cl}$, which represents I_{cl} simultaneous eigendecompositions. The diagonal entries of $\mathbf{\Lambda}_i \mathbf{\Lambda}_0^{-1}$ are $c_{im} / \sum_{j=1}^{I_{cl}} \alpha_j c_{jm}$ for $m = 1, \dots, M$. If $c_{im} \neq 0$ the m -th diagonal entry can be written as $1 / \sum_{j=1}^{I_{cl}} \alpha_j c_{jm} / c_{im}$. We see that this diagonal entry depends on the direction of the m -th column of \mathbf{C} . Thus, the nonzero diagonal entries of $\mathbf{\Lambda}_i \mathbf{\Lambda}_0^{-1}$ are distinct if the corresponding columns in \mathbf{C} have distinct directions. In that case, the columns of $\mathbf{F} = [\mathbf{f}_1 \ \dots \ \mathbf{f}_M]$ corresponding to the nonzero distinct diagonal entries of $\mathbf{\Lambda}_i \mathbf{\Lambda}_0^{-1}$ can be determined up to permutation and scaling from the eigendecomposition of $\mathbf{Y}_i \mathbf{Y}_0^{-1}$. We know that for any m there is some $c_{im} \neq 0$ for $i = 1, \dots, I_{cl}$. Therefore, if the matrix \mathbf{C} has no proportional columns, the matrix \mathbf{F} can be recovered by performing the simultaneous eigendecompositions defined in (3.18).

Let us suppose now that the first two columns in \mathbf{C} are proportional (i. e., $c_{i1} = \beta c_{i2}$ for $i = 1, \dots, I_{cl}$ and some constant β) while all other columns have distinct directions. In this case the first two diagonal entries are identical in all of the matrices $\mathbf{\Lambda}_i \mathbf{\Lambda}_0^{-1}$ for $i = 1, \dots, I_{cl}$. Then the eigendecompositions of $\mathbf{Y}_i \mathbf{Y}_0^{-1}$ for $i = 1, \dots, I_{cl}$ can uniquely recover all the columns of \mathbf{F} except \mathbf{f}_1 and \mathbf{f}_2 . It can recover only $\text{span}\{\mathbf{f}_1, \mathbf{f}_2\}$. In other words, if any two columns in \mathbf{C} are proportional the corresponding two columns in \mathbf{F} cannot be recovered individually and only the subspace spanned by these two columns is uniquely determined. In general, columns of the matrix \mathbf{F} can be recovered up to permutation and scaling from (3.17) if (1) $\text{rank}(\mathbf{F}) = M$ and (2) the matrix \mathbf{C} has no proportional columns. If (1) $\text{rank}(\mathbf{F}) = M$ and (2) the matrix \mathbf{C} has groups of proportional columns, we can recover subspaces spanned by the corresponding groups of columns in \mathbf{F} . When the columns of \mathbf{F} are determined up to permutation and scaling from (3.17) we can find contributions of different signals to the three way array \mathbf{Y} .

These conditions are well known [73] and they are derived here for completeness. This type of the matrix recovery by joint diagonalization of a set of given matrices is the main estimation principle behind important methods for independent component analysis and blind signal separation (see [19], [72] and references in [74]). Let us explain this estimation technique from another point of view. We observe that (3.16) represents decomposition of \mathbf{Y} into M three-way rank-one terms [75] [76]. This decomposition is unique under certain algebraic conditions [77]. The uniqueness conditions from [77] are

more general than the ones derived above. When the uniqueness conditions hold the rank-one terms in (3.16) are *uniquely* determined. In other words, we can uniquely determine terms in the sum in (3.16), which represent contributions of different transmitted signals to the observed three-way array \mathbf{Y} . Thus, the decomposition of \mathbf{Y} into rank-one terms becomes a parameter estimation tool.

We note that $\text{rank}(\mathbf{F}) = M$ implies $J_{fb} \geq M$. In our application this condition can be satisfied if the number of bins J_{fb} used for the estimation of the fourth order spectral slices is sufficiently high. The condition saying that the matrix \mathbf{C} has no groups of proportional columns requires a certain diversity among activity sequences associated with different signals. Since different signals normally have completely different activity sequences, this condition is practically always satisfied in any application. The condition $\text{rank}(\mathbf{F}) = M$ requires a certain form of spectral diversity. This spectral diversity is generated by use of different pulse shapes and transmit center frequencies and by different channel transfer functions between the transmitters and the sensor. The condition $\text{rank}(\mathbf{F}) = M$ can break down if two or more transmitters use identical pulse shapes and the corresponding channel transfer functions are flat in frequency. In this case there are two or more proportional columns in \mathbf{F} and hence, $\text{rank}(\mathbf{F}) < M$. Let us assume now that $\text{rank}(\mathbf{F}) = M - 1$ because there are two identical received pulse shapes and hence, there are two proportional columns in \mathbf{F} . In that case these two signals contribute only one rank one term in the sum in (3.16). These two signals are characterized with one column in \mathbf{F} representing their common pulse shape and one column in \mathbf{C} representing activity sequence for both signals. Therefore, in this case the proposed analysis method is still applicable but now we need to decompose \mathbf{Y} into $M - 1$ rank one terms instead of M rank one terms.

In the previous analysis we used the slice $S_{04x}(f, v) = S_{4x}(f, v, -v)$, which is only one part of the information contained in the fourth order spectrum. Let us now consider the slice $S_{\Delta v 4x}(f, v) = S_{4x}(f, v + \Delta v, -v)$ where Δv is a constant. Let $S_{\Delta v 4}(f, v, i)$ be the fourth order spectrum slice defined above for the i -th cluster. Equation (3.10) now

becomes

$$S_{\Delta v 4}(f, v, i) = \sum_{m=1}^M H_m(f) H_m(f + \Delta v)^* H_m(v + \Delta v) H_m(v)^* S_{\Delta v 4 m}(f, v) c_{im} + S_{\Delta v 4 n}(f, v) \quad (3.19)$$

where $i = 1, \dots, I_{cl}$ and $S_{\Delta v 4 m}(f, v)$ is the fourth order spectrum slice of the m -th transmitted signal. Again, assuming the receiver noise is Gaussian, we have $S_{\Delta v 4 n}(f, v) = 0$.

Let us now define the three-way array $\mathbf{Y}_{\Delta v}$ with entries

$$\mathbf{Y}_{\Delta v}(j, n, i) = \int_{(j-1)\Delta f}^{j\Delta f} \int_{(n-1)\Delta f}^{n\Delta f} S_{\Delta v 4}(f, v, i) df dv = \sum_{m=1}^M q_{\Delta v, jnm} c_{im} \quad (3.20)$$

where $q_{jnm} = \int_{(j-1)\Delta f}^{j\Delta f} \int_{(n-1)\Delta f}^{n\Delta f} H_m(f) H_m(f + \Delta v)^* H_m(v + \Delta v) H_m(v)^* S_{\Delta v 4 m}(f, v) df dv$.

For linear modulation formats we have

$$S_{\Delta v 4 m}(f, v) = k_m P_m(f) P_m(f + \Delta v)^* P_m(v + \Delta v) P_m(v)^* \quad (3.21)$$

which follows from (3.14). It follows that $q_{\Delta v, jnm} = k_m f_{\Delta v, jm} f_{\Delta v, nm}^*$, where $f_{\Delta v, nm} = \int_{(n-1)\Delta f}^{n\Delta f} H_m(f) H_m(f + \Delta v)^* P_m(f) P_m(f + \Delta v)^* df$. In this case the entries of $\mathbf{Y}_{\Delta v}$ are

$$\mathbf{Y}_{\Delta v}(j, n, i) = \sum_{m=1}^M k_m f_{\Delta v, jm} f_{\Delta v, nm}^* c_{im}. \quad (3.22)$$

Now we observe that

$$\mathbf{Y}_{\Delta v, i} = \mathbf{F}_{\Delta v} \mathbf{\Lambda}_i \mathbf{F}_{\Delta v}^H \quad (3.23)$$

where $\mathbf{Y}_{\Delta v, i}$ is the i -th slice of $\mathbf{Y}_{\Delta v}$, $\mathbf{F}_{\Delta v} = [f_{\Delta v, jm}]$ and $\mathbf{\Lambda}_i = \text{diag}([k_1 c_{i1} \dots k_M c_{iM}])$.

We see that the matrix $\mathbf{F}_{\Delta v}$ jointly diagonalizes the matrices $\mathbf{Y}_{\Delta v, i}$ for $i = 1, \dots, I_{cl}$.

Thus, under similar algebraic conditions to those stated above, it is possible to recover

$\mathbf{F}_{\Delta v}$ from $\mathbf{Y}_{\Delta v}$. The entries of $\mathbf{F}_{\Delta v}$ contain information on the phase of $P_m(f)$ and

$H_m(f)$. On the other hand, the entries of \mathbf{F} are functions of $|P_m(f)|$ and $|H_m(f)|$

and hence, they are insensitive to the phase of $P_m(f)$ and $H_m(f)$. We see that the

slices $S_{\Delta v 4 x}(f, v) = S_{4x}(f, v + \Delta v, -v)$ for $\Delta v \neq 0$ also contain useful information and

can be used for signal analysis. By using these slices for different values of Δv it is

possible to exploit full information from the fourth order spectrum. Numerical algo-

rithms presented here can be used for computing the decomposition defined in (3.16).

Computing the decomposition defined in (3.22) for $\Delta v \neq 0$ requires different numerical

algorithms since some quantities in (3.22) are complex. We leave this generalization of the algorithm for future work.

Nonlinear modulation formats. It is not possible to apply the proposed methodology to nonlinear modulation formats in general. Obviously, nonlinearity is not a property and therefore, it cannot be treated without introducing additional assumptions. The proposed analysis method can be generalized when the transmitted signals can be represented or approximated as finite sums of linearly modulated signals. Let us assume that the m -th transmitted signal can be represented as a sum of R_m statistically independent signals in linear format given by (3.13). Then its fourth order spectrum slice can be represented as

$$\mathbf{Q}_m = \sum_{r=1}^{R_m} k_{rm} \mathbf{f}_r^{(m)} \mathbf{f}_r^{(m)T} = \mathbf{F}_m \mathbf{D}_m \mathbf{F}_m^T \quad (3.24)$$

where $\mathbf{D}_m = \text{diag}([k_{1m} \ \dots \ k_{R_m,m}])$ contains constants defined in (3.15) and $\mathbf{F}_m = [\mathbf{f}_1^{(m)} \ \dots \ \mathbf{f}_{R_m}^{(m)}]$ contains the associated squared magnitude frequency responses of the linear signals used to represent the m -th source signal. The entries of \mathbf{Y} can be written as

$$\mathbf{Y}(j, n, i) = \sum_{m=1}^M q_{jnm} c_{im} = \sum_{m=1}^M \sum_{r=1}^{R_m} k_{rm} f_{jrm} f_{nrm} c_{im} \quad (3.25)$$

where $\mathbf{F}_m(j, r) = f_{jrm}$. The matrix slices of the three way array are now

$$\mathbf{Y}_i = [\mathbf{F}_1 \ \dots \ \mathbf{F}_M] \begin{bmatrix} \mathbf{\Lambda}_{i1} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{\Lambda}_{iM} \end{bmatrix} [\mathbf{F}_1 \ \dots \ \mathbf{F}_M]^T \quad (3.26)$$

where $\mathbf{\Lambda}_{im} = \text{diag}([k_{1m} c_{im} \ \dots \ k_{R_m,1} c_{im}])$ for $m = 1, \dots, M$ and $i = 1, \dots, I_{cl}$. Now we need to assume that $J_{fb} \geq R_1 + \dots + R_M$ and $\text{rank}([\mathbf{F}_1 \ \dots \ \mathbf{F}_M]) = R_1 + \dots + R_M$. Again, we can assume $J_{fb} = R_1 + \dots + R_M$ and define $\mathbf{Y}_0 = \mathbf{F} \mathbf{\Lambda}_0 \mathbf{F}^T$

$$\mathbf{Y}_0 = \sum_{i=1}^{I_{cl}} \alpha_i \mathbf{Y}_i = [\mathbf{F}_1 \ \dots \ \mathbf{F}_M] \begin{bmatrix} \mathbf{\Lambda}_{01} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{\Lambda}_{0M} \end{bmatrix} [\mathbf{F}_1 \ \dots \ \mathbf{F}_M]^T \quad (3.27)$$

where $\mathbf{\Lambda}_{0m} = \sum_{i=1}^{I_{cl}} \alpha_i \mathbf{\Lambda}_{im}$ for $m = 1, \dots, M$. Again, we can pick the vector $\begin{bmatrix} \alpha_1 & \dots & \alpha_{I_{cl}} \end{bmatrix}$ so that $\mathbf{\Lambda}_{0m}$ for $m = 1, \dots, M$ are invertible and form

$$\mathbf{Y}_i \mathbf{Y}_0^{-1} = \begin{bmatrix} \mathbf{F}_1 & \dots & \mathbf{F}_M \end{bmatrix} \begin{bmatrix} \mathbf{\Lambda}_{i1} \mathbf{\Lambda}_{01}^{-1} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{\Lambda}_{iM} \mathbf{\Lambda}_{0M}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{F}_1 & \dots & \mathbf{F}_M \end{bmatrix}^{-1} \quad (3.28)$$

for $i = 1, \dots, I_{cl}$. Again, we have I_{cl} simultaneous eigendecompositions, but now there are M groups of repeated diagonal entries in (3.28). These repeated entries appear at exactly the same positions for all matrices $\mathbf{Y}_i \mathbf{Y}_0^{-1}$ $i = 1, \dots, I_{cl}$. Assuming that \mathbf{C} has no proportional columns from (3.28) we can recover $\text{span}\{\mathbf{F}_m\}$ for $m = 1, \dots, M$. This conclusion follows from the previous analysis of the joint diagonalization problem. More precisely, from (3.28) we cannot find \mathbf{F}_m but we can find $\tilde{\mathbf{F}}_m = \mathbf{F}_m \mathbf{T}_m$ where \mathbf{T}_m is an invertible matrix. This indeterminacy does not prevent us from estimating contributions of different signals to the observed three way array \mathbf{Y} since

$$\mathbf{Y}_i = \begin{bmatrix} \tilde{\mathbf{F}}_1 & \dots & \tilde{\mathbf{F}}_M \end{bmatrix} \begin{bmatrix} \mathbf{T}_1^{-1} \mathbf{\Lambda}_{i1} \mathbf{T}_1^{T-1} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{T}_M^{-1} \mathbf{\Lambda}_{iM} \mathbf{T}_M^{T-1} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{F}}_1 & \dots & \tilde{\mathbf{F}}_M \end{bmatrix}^T \quad (3.29)$$

where $m = 1, \dots, M$ and $i = 1, \dots, I_{cl}$.

The problem of recovering $\begin{bmatrix} \mathbf{F}_1 & \dots & \mathbf{F}_M \end{bmatrix}$ from the matrices \mathbf{Y}_i given by (3.26) is called joint block diagonalization and it arises in some signal processing problems [78] [79]. Again, we can explain the method from another point of view. Equation (3.25) represents decomposition of \mathbf{Y} into M block terms [80]. Now each signal contributes one block term rather than one rank-one term, which was the case with linear modulation formats. The decomposition into block terms can also be unique under certain algebraic conditions [80], which are generalization of those from [77]. When these conditions hold block terms in (3.25), representing contributions of different transmitted signals to \mathbf{Y} , are *uniquely* determined and hence, the decomposition of \mathbf{Y} into block terms becomes a parameter estimation tool.

definition in [81], $\text{rank}(\begin{bmatrix} \mathbf{F}_1 & \dots & \mathbf{F}_M \end{bmatrix}) = R_1 + \dots + R_M$ implies that the subspaces $\text{col}\{\mathbf{F}_m\}$ for $m = 1, \dots, M$ are linearly independent. Thus, this condition requires a certain form of spectral diversity among the received signals. The sources of this diversity are use of different transmitted signal formats and different center frequencies and propagation of signals through different channels between the transmitters and the sensor. If there are two signals with proportional fourth order spectrum slices (i. e., $\mathbf{Q}_1 = \alpha \mathbf{Q}_2$ where α is a constant) the proposed method should be used with $M - 1$ instead of M .

The condition requiring that the subspaces $\text{col}\{\mathbf{F}_m\}$ for $m = 1, \dots, M$ are linearly independent can break down if for example, $\text{rank}(\begin{bmatrix} \mathbf{F}_1 & \mathbf{F}_2 \end{bmatrix}) < R_1 + R_2$. In order to understand what happens then we consider the more general model of the form (3.30) with

$$\mathbf{F} = \begin{bmatrix} \mathbf{f}_1 & \dots & \mathbf{f}_R \end{bmatrix}, \quad (3.33)$$

$$\mathbf{F}_V = \begin{bmatrix} \text{vecr}(\mathbf{f}_1 \mathbf{f}_1^T) & \dots & \text{vecr}(\mathbf{f}_R \mathbf{f}_R^T) \end{bmatrix}, \quad (3.34)$$

$$\mathbf{C} = \begin{bmatrix} \mathbf{c}_1 & \dots & \mathbf{c}_M \end{bmatrix} \quad (3.35)$$

and the R by M matrix $\mathbf{K} = [k_{rm}]$ which is not necessarily constrained to the block form given in (3.32). The entries of the three way array \mathbf{Y} are

$$\mathbf{Y}(j, n, i) = \sum_{m=1}^M \sum_{r=1}^R k_{rm} f_{jr} f_{nr} c_{im}. \quad (3.36)$$

The matrix slices of \mathbf{Y} are now

$$\mathbf{Y}_i = \begin{bmatrix} \mathbf{f}_1 & \dots & \mathbf{f}_R \end{bmatrix} \begin{bmatrix} \sum_{m=1}^M k_{1m} c_{im} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sum_{m=1}^M k_{Rm} c_{im} \end{bmatrix} \begin{bmatrix} \mathbf{f}_1 & \dots & \mathbf{f}_R \end{bmatrix}^T \quad (3.37)$$

for $i = 1, \dots, I_{cl}$. Again, we can pick the vector $\begin{bmatrix} \alpha_1 & \dots & \alpha_{I_{cl}} \end{bmatrix}$ such that the matrix $\mathbf{Y}_0 = \sum_{i=1}^{I_{cl}} \alpha_i \mathbf{Y}_i$ is invertible and form

$$\mathbf{Y}_i \mathbf{Y}_0^{-1} = \mathbf{F} \begin{bmatrix} \frac{\sum_{m=1}^M k_{1m} c_{im}}{\sum_{i=1}^{I_{cl}} \sum_{m=1}^M \alpha_i k_{1m} c_{im}} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \frac{\sum_{m=1}^M k_{Rm} c_{im}}{\sum_{i=1}^{I_{cl}} \sum_{m=1}^M \alpha_i k_{Rm} c_{im}} \end{bmatrix} \mathbf{F}^{-1} \quad (3.38)$$

where we assumed that $J_{fb} = R$ and that \mathbf{F} is invertible. Again, we obtain the simultaneous eigendecompositions. Let us now consider several concrete examples.

In the first example we consider the case where

$$\mathbf{K} = \begin{bmatrix} k_{11} & 0 \\ k_{21} & 0 \\ k_{31} & k_{12} \\ k_{41} & k_{22} \\ 0 & k_{32} \\ 0 & k_{42} \end{bmatrix}. \quad (3.39)$$

We obtain the following eigendecompositions

$$\mathbf{Y}_i \mathbf{Y}_0^{-1} = \mathbf{F} \text{diag}(\mathbf{\Lambda}(i, :)) \mathbf{F}^{-1} \quad (3.40)$$

where $i = 1, \dots, I_{cl}$ and $\mathbf{\Lambda}(i, :)$ is the i -th column of the matrix

$$\mathbf{\Lambda} = \begin{bmatrix} \frac{\mathbf{c}_1^T}{\sum_{i=1}^{I_{cl}} \alpha_i c_{i1}} \\ \frac{\mathbf{c}_1^T}{\sum_{i=1}^{I_{cl}} \alpha_i c_{i1}} \\ \frac{k_{31} \mathbf{c}_1^T + k_{12} \mathbf{c}_2^T}{\sum_{i=1}^{I_{cl}} \alpha_i (k_{31} c_{i1} + k_{12} c_{i2})} \\ \frac{k_{41} \mathbf{c}_1^T + k_{22} \mathbf{c}_2^T}{\sum_{i=1}^{I_{cl}} \alpha_i (k_{41} c_{i1} + k_{12} c_{i2})} \\ \frac{\mathbf{c}_2^T}{\sum_{i=1}^{I_{cl}} \alpha_i c_{i2}} \\ \frac{\mathbf{c}_2^T}{\sum_{i=1}^{I_{cl}} \alpha_i c_{i2}} \end{bmatrix}. \quad (3.41)$$

The matrix $\mathbf{\Lambda}$ can be recovered by solving the eigendecompositions from (3.40). If \mathbf{C} is invertible the contributions of the two signals can be resolved if we can find the directions of \mathbf{c}_1 and \mathbf{c}_2 . We see that $\text{col}(\mathbf{\Lambda}^T) = \text{span}\{\mathbf{c}_1, \mathbf{c}_2\}$. The top two rows in $\mathbf{\Lambda}$ have the same direction as \mathbf{c}_1 and the bottom two rows in $\mathbf{\Lambda}$ have the same direction as \mathbf{c}_2 . Assuming that there are no other subsets of the rows of $\mathbf{\Lambda}$ which contain proportional vectors it is possible to identify the directions of \mathbf{c}_1 and \mathbf{c}_2 from $\mathbf{\Lambda}$.

In the second example we consider the case where

$$\mathbf{K} = \begin{bmatrix} k_{11} & 0 \\ k_{21} & k_{12} \\ k_{31} & k_{22} \\ k_{41} & k_{32} \\ k_{51} & k_{42} \\ 0 & k_{52} \end{bmatrix}. \quad (3.42)$$

We obtain the following eigendecompositions

$$\mathbf{Y}_i \mathbf{Y}_0^{-1} = \mathbf{F} \text{diag}(\boldsymbol{\Lambda}(i, :)) \mathbf{F}^{-1} \quad (3.43)$$

where $i = 1, \dots, I_{cl}$ and $\boldsymbol{\Lambda}(i, :)$ is the i -th column of the matrix

$$\boldsymbol{\Lambda} = \begin{bmatrix} \frac{\mathbf{c}_1^T}{\sum_{i=1}^{I_{cl}} \alpha_i c_{i1}} \\ \frac{k_{21} \mathbf{c}_1^T + k_{12} \mathbf{c}_2^T}{\sum_{i=1}^{I_{cl}} \alpha_i (k_{21} c_{i1} + k_{12} c_{i2})} \\ \frac{k_{31} \mathbf{c}_1^T + k_{22} \mathbf{c}_2^T}{\sum_{i=1}^{I_{cl}} \alpha_i (k_{31} c_{i1} + k_{22} c_{i2})} \\ \frac{k_{41} \mathbf{c}_1^T + k_{32} \mathbf{c}_2^T}{\sum_{i=1}^{I_{cl}} \alpha_i (k_{41} c_{i1} + k_{32} c_{i2})} \\ \frac{k_{51} \mathbf{c}_1^T + k_{42} \mathbf{c}_2^T}{\sum_{i=1}^{I_{cl}} \alpha_i (k_{51} c_{i1} + k_{42} c_{i2})} \\ \frac{\mathbf{c}_2^T}{\sum_{i=1}^{I_{cl}} \alpha_i c_{i2}} \end{bmatrix}. \quad (3.44)$$

In this case the rows of $\boldsymbol{\Lambda}$ representing the directions of \mathbf{c}_1 and \mathbf{c}_2 are not repeated and hence they cannot be identified from $\boldsymbol{\Lambda}$. In this example we can identify only $\text{span}\{\mathbf{c}_1, \mathbf{c}_2\} = \text{col}(\boldsymbol{\Lambda}^T)$.

From these two examples we can draw the following conclusion. If the subspaces of the two signals partially overlap their contributions to \mathbf{Y} can be recovered if the size of the each nonoverlapping portion is at least two. In the first example this condition is satisfied and hence, we can recover the directions of \mathbf{c}_1 and \mathbf{c}_2 . Assuming that \mathbf{C} is of full rank and using the recovered directions of \mathbf{c}_1 and \mathbf{c}_2 we can find the contributions of the two signals to \mathbf{Y} . In the second example we can find only $\text{span}\{\mathbf{c}_1, \mathbf{c}_2\}$. In order to recover the contributions of the two signals to \mathbf{Y} we need to make additional assumptions. For example, this recovery is possible if the two signals transmit nonoverlapping signals in time.

In the third example we consider the case where

$$\mathbf{K} = \begin{bmatrix} k_{11} & k_{12} & 0 & 0 \\ k_{21} & k_{22} & 0 & 0 \\ k_{31} & k_{32} & 0 & 0 \\ k_{41} & k_{42} & k_{13} & k_{14} \\ k_{51} & k_{52} & k_{23} & k_{24} \\ k_{61} & k_{62} & k_{33} & k_{34} \\ 0 & 0 & k_{43} & k_{44} \\ 0 & 0 & k_{53} & k_{54} \\ 0 & 0 & k_{63} & k_{64} \end{bmatrix}. \quad (3.45)$$

We obtain the following eigendecompositions

$$\mathbf{Y}_i \mathbf{Y}_0^{-1} = \mathbf{F} \text{diag}(\boldsymbol{\Lambda}(i, :)) \mathbf{F}^{-1} \quad (3.46)$$

where $i = 1, \dots, I_{cl}$ and $\boldsymbol{\Lambda}(i, :)$ is the i -th column of the matrix

$$\boldsymbol{\Lambda} = \begin{bmatrix} \frac{k_{11}\mathbf{c}_1^T + k_{12}\mathbf{c}_2^T}{\sum_{i=1}^{I_{cl}} \alpha_i(k_{11}c_{i1} + k_{12}c_{i2})} \\ \frac{k_{21}\mathbf{c}_1^T + k_{22}\mathbf{c}_2^T}{\sum_{i=1}^{I_{cl}} \alpha_i(k_{21}c_{i1} + k_{22}c_{i2})} \\ \frac{k_{31}\mathbf{c}_1^T + k_{32}\mathbf{c}_2^T}{\sum_{i=1}^{I_{cl}} \alpha_i(k_{31}c_{i1} + k_{32}c_{i2})} \\ \frac{k_{41}\mathbf{c}_1^T + k_{42}\mathbf{c}_2^T + k_{13}\mathbf{c}_3^T + k_{14}\mathbf{c}_4^T}{\sum_{i=1}^{I_{cl}} \alpha_i(k_{41}c_{i1} + k_{42}c_{i2} + k_{13}c_{i3} + k_{14}c_{i4})} \\ \frac{k_{51}\mathbf{c}_1^T + k_{52}\mathbf{c}_2^T + k_{23}\mathbf{c}_3^T + k_{24}\mathbf{c}_4^T}{\sum_{i=1}^{I_{cl}} \alpha_i(k_{51}c_{i1} + k_{52}c_{i2} + k_{23}c_{i3} + k_{24}c_{i4})} \\ \frac{k_{61}\mathbf{c}_1^T + k_{62}\mathbf{c}_2^T + k_{33}\mathbf{c}_3^T + k_{34}\mathbf{c}_4^T}{\sum_{i=1}^{I_{cl}} \alpha_i(k_{61}c_{i1} + k_{62}c_{i2} + k_{33}c_{i3} + k_{34}c_{i4})} \\ \frac{k_{43}\mathbf{c}_3^T + k_{44}\mathbf{c}_4^T}{\sum_{i=1}^{I_{cl}} \alpha_i(k_{43}c_{i3} + k_{44}c_{i4})} \\ \frac{k_{53}\mathbf{c}_3^T + k_{54}\mathbf{c}_4^T}{\sum_{i=1}^{I_{cl}} \alpha_i(k_{53}c_{i3} + k_{54}c_{i4})} \\ \frac{k_{63}\mathbf{c}_3^T + k_{64}\mathbf{c}_4^T}{\sum_{i=1}^{I_{cl}} \alpha_i(k_{63}c_{i3} + k_{64}c_{i4})} \end{bmatrix}. \quad (3.47)$$

We see that the top three vectors define a rank two matrix whose row space is $\text{span}\{\mathbf{c}_1, \mathbf{c}_2\}$.

Also, we see that the bottom three vectors define a rank two matrix whose row space is $\text{span}\{\mathbf{c}_3, \mathbf{c}_4\}$. Assuming any other submatrix containing three rows of $\boldsymbol{\Lambda}$ is of the rank three it is possible to identify $\text{span}\{\mathbf{c}_1, \mathbf{c}_2\}$ and $\text{span}\{\mathbf{c}_3, \mathbf{c}_4\}$ from $\boldsymbol{\Lambda}$. However, it is not possible to identify the directions of \mathbf{c}_1 and \mathbf{c}_2 from $\text{span}\{\mathbf{c}_1, \mathbf{c}_2\}$ because the subspaces

used by these two signals completely overlap. Similarly, it is not possible to identify the directions of \mathbf{c}_3 and \mathbf{c}_4 from $\text{span}\{\mathbf{c}_3, \mathbf{c}_4\}$. The problem of identifying subspaces in a set of vectors appears also in [32] and [82].

In the fourth example we consider the case where

$$\mathbf{K} = \begin{bmatrix} k_{11} & k_{12} & 0 & 0 \\ k_{21} & k_{22} & 0 & 0 \\ k_{31} & k_{32} & k_{13} & k_{14} \\ k_{41} & k_{42} & k_{23} & k_{24} \\ k_{51} & k_{52} & k_{33} & k_{34} \\ k_{61} & k_{62} & k_{43} & k_{44} \\ k_{71} & k_{72} & k_{53} & k_{54} \\ 0 & 0 & k_{63} & k_{64} \\ 0 & 0 & k_{73} & k_{74} \end{bmatrix}. \quad (3.48)$$

We obtain the following eigendecompositions

$$\mathbf{Y}_i \mathbf{Y}_0^{-1} = \mathbf{F} \text{diag}(\boldsymbol{\Lambda}(i, :)) \mathbf{F}^{-1} \quad (3.49)$$

where $i = 1, \dots, I_{cl}$ and $\boldsymbol{\Lambda}(i, :)$ is the i -th column of the matrix

$$\boldsymbol{\Lambda} = \begin{bmatrix} \frac{k_{11}\mathbf{c}_1^T + k_{12}\mathbf{c}_2^T}{\sum_{i=1}^{I_{cl}} \alpha_i(k_{11}c_{i1} + k_{12}c_{i2})} \\ \frac{k_{21}\mathbf{c}_1^T + k_{22}\mathbf{c}_2^T}{\sum_{i=1}^{I_{cl}} \alpha_i(k_{21}c_{i1} + k_{22}c_{i2})} \\ \frac{k_{31}\mathbf{c}_1^T + k_{32}\mathbf{c}_2^T + k_{13}\mathbf{c}_3^T + k_{14}\mathbf{c}_4^T}{\sum_{i=1}^{I_{cl}} \alpha_i(k_{31}c_{i1} + k_{32}c_{i2} + k_{13}c_{i3} + k_{14}c_{i4})} \\ \frac{k_{41}\mathbf{c}_1^T + k_{42}\mathbf{c}_2^T + k_{23}\mathbf{c}_3^T + k_{24}\mathbf{c}_4^T}{\sum_{i=1}^{I_{cl}} \alpha_i(k_{41}c_{i1} + k_{42}c_{i2} + k_{23}c_{i3} + k_{24}c_{i4})} \\ \frac{k_{51}\mathbf{c}_1^T + k_{52}\mathbf{c}_2^T + k_{33}\mathbf{c}_3^T + k_{34}\mathbf{c}_4^T}{\sum_{i=1}^{I_{cl}} \alpha_i(k_{51}c_{i1} + k_{52}c_{i2} + k_{33}c_{i3} + k_{34}c_{i4})} \\ \frac{k_{61}\mathbf{c}_1^T + k_{62}\mathbf{c}_2^T + k_{43}\mathbf{c}_3^T + k_{44}\mathbf{c}_4^T}{\sum_{i=1}^{I_{cl}} \alpha_i(k_{61}c_{i1} + k_{62}c_{i2} + k_{43}c_{i3} + k_{44}c_{i4})} \\ \frac{k_{71}\mathbf{c}_1^T + k_{72}\mathbf{c}_2^T + k_{53}\mathbf{c}_3^T + k_{54}\mathbf{c}_4^T}{\sum_{i=1}^{I_{cl}} \alpha_i(k_{71}c_{i1} + k_{72}c_{i2} + k_{53}c_{i3} + k_{54}c_{i4})} \\ \frac{k_{63}\mathbf{c}_3^T + k_{64}\mathbf{c}_4^T}{\sum_{i=1}^{I_{cl}} \alpha_i(k_{63}c_{i3} + k_{64}c_{i4})} \\ \frac{k_{73}\mathbf{c}_3^T + k_{74}\mathbf{c}_4^T}{\sum_{i=1}^{I_{cl}} \alpha_i(k_{73}c_{i3} + k_{74}c_{i4})} \end{bmatrix}. \quad (3.50)$$

In this example we cannot find appropriate submatrices of $\boldsymbol{\Lambda}$ to identify $\text{span}\{\mathbf{c}_1, \mathbf{c}_2\}$ and $\text{span}\{\mathbf{c}_3, \mathbf{c}_4\}$. The top two rows of $\boldsymbol{\Lambda}$ define $\text{span}\{\mathbf{c}_1, \mathbf{c}_2\}$ and the bottom two rows

define $\text{span}\{\mathbf{c}_3, \mathbf{c}_4\}$. However, any other submatrix containing two rows of \mathbf{A} is also of the rank two and hence, there is no way to select the submatrices for identification of $\text{span}\{\mathbf{c}_1, \mathbf{c}_2\}$ and $\text{span}\{\mathbf{c}_3, \mathbf{c}_4\}$. Therefore, in this example we can identify only $\text{span}\{\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4\}$. Main conclusion from the last two examples is that in order to resolve $\text{span}\{\mathbf{c}_1, \mathbf{c}_2\}$ and $\text{span}\{\mathbf{c}_3, \mathbf{c}_4\}$ the size of the nonoverlapping portions of their subspaces must be at least three. This conclusion can be easily generalized. For example, if there were two groups of signals where each group has three signals using the same subspace then in order to resolve $\text{span}\{\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3\}$ and $\text{span}\{\mathbf{c}_4, \mathbf{c}_5, \mathbf{c}_6\}$ the size of the nonoverlapping portions of their subspaces must be at least four and so on.

In order to find contributions of different signals to \mathbf{Y} we need to recover the matrices \mathbf{F} , \mathbf{K} , and \mathbf{C} from \mathbf{Y} whose entries are given by (3.36). In order to do this we need to know the parameters M and R . Let us describe how (at least in theory) the parameters M and R can be determined from \mathbf{Y} . First, we apply vecr operator on (3.12) for $i = 1, \dots, I_{cl}$. We have

$$\begin{bmatrix} \text{vecr}(\mathbf{Y}_1) & \dots & \text{vecr}(\mathbf{Y}_{I_{cl}}) \end{bmatrix} = \mathbf{Q}_V \mathbf{C}^T \quad (3.51)$$

where

$$\mathbf{Q}_V = \begin{bmatrix} \text{vecr}(\mathbf{Q}_1) & \dots & \text{vecr}(\mathbf{Q}_M) \end{bmatrix}. \quad (3.52)$$

Assuming that $J_{fb} + \binom{J_{fb}}{2} \geq M$, $I_{cl} \geq M$, and that \mathbf{Q}_V and \mathbf{C} are of full rank we have

$$\text{rank} \left(\begin{bmatrix} \text{vecr}(\mathbf{Y}_1) & \dots & \text{vecr}(\mathbf{Y}_{I_{cl}}) \end{bmatrix} \right) = M \quad (3.53)$$

which can be used to determine M . Next, we show how R can be found from \mathbf{Y} . We rewrite (3.37) as

$$\mathbf{Y}_i = \mathbf{F} \text{diag}(\mathbf{L}(i, :)) \mathbf{F}^T \quad (3.54)$$

where $\mathbf{L}(i, :)$ is the i -th row of the matrix

$$\mathbf{L} = \begin{bmatrix} \mathbf{l}_1 & \dots & \mathbf{l}_R \end{bmatrix} = \mathbf{C} \mathbf{K}^T. \quad (3.55)$$

It is very simple to check that

$$\begin{bmatrix} \mathbf{Y}_1 & \dots & \mathbf{Y}_{I_{cl}} \end{bmatrix} = \mathbf{F} (\mathbf{L} \odot \mathbf{F})^T \quad (3.56)$$

where the symbol \odot denotes the Khatri-Rao product defined as [76]

$$\mathbf{L} \odot \mathbf{F} = \begin{bmatrix} \mathbf{l}_1 \otimes \mathbf{f}_1 & \dots & \mathbf{l}_R \otimes \mathbf{f}_R \end{bmatrix} \quad (3.57)$$

where the symbol \otimes denotes the Kronecker product [83]. Assuming that $J_{fb} \geq R$, $\text{rank}(\mathbf{F}) = R$, and $\text{rank}(\mathbf{L} \odot \mathbf{F}) = R$ we have

$$\text{rank} \left(\begin{bmatrix} \mathbf{Y}_1 & \dots & \mathbf{Y}_{I_{cl}} \end{bmatrix} \right) = R \quad (3.58)$$

which can be used to determine R .

The final problem in recovering the matrices \mathbf{F} , \mathbf{K} , and \mathbf{C} from \mathbf{Y} is that the structure of the matrix \mathbf{K} is not known in advance. For example, if the matrix \mathbf{K} is of the form given in (3.32) then in recovering \mathbf{F} , \mathbf{K} , and \mathbf{C} from \mathbf{Y} the matrix \mathbf{K} has to be appropriately constrained to obtain the desired solution. In other words, we need to know which entries of \mathbf{K} are not equal to zeros. This is crucial since we have seen that the number and positions of nonzero and zero entries determine if the contributions of different signals can be recovered from \mathbf{Y} . In general, the contributions of different signals can be recovered from \mathbf{Y} if the matrix \mathbf{K} is sufficiently sparse. As we explained earlier the recovery is possible for different sparsity patterns of the matrix \mathbf{K} . The main problem here is that the sparsity pattern of the matrix \mathbf{K} is not known in advance. This problem is overcome by observing that the desired decomposition of \mathbf{Y} is always obtained for the matrix \mathbf{K} which has the smallest possible number of nonzero entries. Therefore, in recovering the matrices \mathbf{F} , \mathbf{K} , and \mathbf{C} from \mathbf{Y} we obtain the desired solution when the matrix \mathbf{K} is constrained to be maximally sparse.

Since we assumed Gaussian sensor noise, there is no noise contribution to \mathbf{Y} . We can extend our approach to any stationary non-Gaussian noise that can be represented or approximated as a sum of finite number of linear signals. Such noise would contribute its own block term to \mathbf{Y} . Since the noise is stationary the corresponding column in \mathbf{C} would consist of all ones.

Let us now consider the system model where we use only second order spectrum. Let $S_2(f, i)$ be the second order spectrum for the i -th cluster. Assuming statistically

independent signals over each segment we have

$$S_2(f, i) = \sum_{m=1}^M |H_m(f)|^2 S_{2m}(f) c_{im} + S_{2n}(f) \quad (3.59)$$

where $S_{2m}(f)$ is the second order spectrum of the m -th transmitted signal, $S_{2n}(f)$ is the second order spectrum of the sensor noise, and $c_{im} \in \{0, 1\}$ indicates whether the m -th source is on during the segments of the i -th cluster for $i = 1, \dots, I_{cl}$. Let us define the matrix \mathbf{Z} with entries

$$\mathbf{Z}(j, i) = \int_{(j-1)\Delta f}^{j\Delta f} S_2(f, i) df = \sum_{m=1}^M g_{jm} c_{im} + n_j \quad (3.60)$$

where $g_{jm} = \int_{(j-1)\Delta f}^{j\Delta f} |H_m(f)|^2 S_{2m}(f) df$ and $n_j = \int_{(j-1)\Delta f}^{j\Delta f} S_{2n}(f) df$ for $j = 1, \dots, J_{fb}$.

Finally, our system model becomes

$$\mathbf{Z} = \begin{bmatrix} \mathbf{G} & \mathbf{n} \end{bmatrix} \begin{bmatrix} \mathbf{C} & \mathbf{1} \end{bmatrix}^T \quad (3.61)$$

where $\mathbf{G} = [g_{jm}]$, $\mathbf{n} = [n_j]$, $\mathbf{C} = [c_{im}]$ and $\mathbf{1}$ is a vector of the length I_{cl} containing all ones. In order to find the contributions of different signals and noise to \mathbf{Z} we need to perform the matrix factorization from (3.61). In general, matrix factorizations are not unique. In other words, for a given \mathbf{Z} the matrices \mathbf{G} , \mathbf{n} , and \mathbf{C} cannot be uniquely determined without imposing additional constraints on them. In contrast to matrix factorizations, factorizations of three-way arrays are often unique and hence, they can be used as parameter estimation tools [76] [84]. These facts explain why we had to use fourth order spectrum in our application. Finally, we note that, if the matrix \mathbf{C} has been estimated from the decomposition of \mathbf{Y} and if $\text{rank}(\begin{bmatrix} \mathbf{C} & \mathbf{1} \end{bmatrix}) = M + 1$, we can estimate \mathbf{G} and \mathbf{n} from (3.61).

3.2 Estimation of fourth order spectra

First step in the application of the proposed signal analysis method is estimation of the fourth order spectrum slices defined in (3.9) from the received signal samples. There are numerous methods for estimation of second order spectrum (see [34], [35] and references therein). Methods for second order spectrum estimation of stationary signals have been extensively studied and their properties are well understood. On the other hand,

methods for estimation of higher order spectra have been much less studied. Methods for estimation of third order spectrum can be found in [85] [86] [69]. Methods for estimation of fourth order spectrum are discussed in [36], [38] and references therein. As pointed out in [69] methods for estimation of higher order spectra are not well understood. As will be seen, estimation of the fourth order spectrum slices in our application has some unique aspects and it is not straightforward. For these reasons we will give a detailed description of the estimation procedure.

We wish to estimate the fourth order spectrum slices defined in (3.9). The estimation will be done on the discrete time signal $r(k)$ for $k = 1, \dots, K$ obtained by sampling $r(t)$ with the sampling period T_s . Each fourth order spectrum slice will be estimated from the subset of samples of $r(k)$ generated by the same combination of active transmitted signals. In general, this subset consists of several continuous time segments. We assume that these segments have been localized in time using the proposed segmentation algorithm. Let us first describe how to estimate the fourth order spectrum slice on one such continuous time segment with a fixed combination of active transmitters. Let the samples on this segment be $r(k)$ for $k = 1, \dots, K_s$ where $K_s T_s$ is the duration of the segment. Before proceeding with the estimation procedure we need to discuss some important matters concerning the statistics of the signals $s_m(t)$ from (2.1).

As already mentioned most communication signals are cyclostationary, which means that their statistics are periodic functions of time. This periodic dependency is caused by periodically repetitive signal generation mechanisms (e.g., periodic output of information bearing waveforms, multiplication with periodic carriers, periodic spreading, periodic hopping, etc.) [11]. Let $s_1(t)$ be persistently active for $0 \leq t \leq K_s T_s$ and let the modulation format of this signal be associated with period T_1 . Let us define the second order statistic $C_{2s_1}(t, \tau) = E\{s_1(t + \tau)s_1(t)^*\}$. If $C_{2s_1}(t, \tau) = C_{2s_1}(t + T_1, \tau)$ the signal $s_1(t)$ is second order cyclostationary. Let us consider the fourth order statistic $C_{4s_1}(t, \tau_1, \tau_2, \tau_3) = cum\{s_1(t + \tau_1), s_1(t + \tau_2), s_1(t + \tau_3)^*, s_1(t)^*\}$, where the cumulant operator is defined in [36]. If $C_{4s_1}(t, \tau_1, \tau_2, \tau_3) = C_{4s_1}(t + T_1, \tau_1, \tau_2, \tau_3)$ the signal $s_1(t)$ is fourth order cyclostationary and so on. In general, when we say that a signal is cyclostationary it means that some of its statistics vary periodically with time. The

signal $s_1(t)$ can be cyclostationary for $0 \leq t \leq K_s T_s$ only if it is persistently active on that time interval as we assumed. If during this time interval $s_1(t)$ has silent periods or multiple packet transmissions, this signal cannot be cyclostationary on the time interval because any breaks in signal transmission disrupt periodic variation of statistics in time.

Since the received signal $r(t)$ is sampled with T_s it is important to understand relationship between the statistics of $s_1(t)$ and those of the discrete time signal $s_1(k)$ obtained by sampling $s_1(t)$ with T_s . Let us first consider the case where $T_1 = N_1 T_s$ for some integer N_1 . If $N_1 \geq 2$ the signal $s_1(k)$ is cyclostationary with the period N_1 . If $N_1 = 1$ statistics of $s_1(k)$ become independent of time and hence, $s_1(k)$ is a stationary signal. If the relation $T_1 = N_1 T_s$ does not hold for any integer N_1 the signal $s_1(k)$ is either cyclostationary with some period larger than N_1 or not cyclostationary at all. Therefore, the statistics of $s_1(t)$ are accurately reflected in the statistics of $s_1(k)$ if $T_1 = N_1 T_s$ where N_1 is an integer and $N_1 \geq 2$. Let us assume now that the signals $s_m(t)$ for $m = 1, \dots, M$ are cyclostationary with periods T_m for $m = 1, \dots, M$. We proceed under the assumption that the sampling period T_s is chosen so that $T_m = N_m T_s$ where N_m is an integer and $N_m \geq 2$ for $m = 1, \dots, M$.

We now return to the problem of estimation of the fourth order spectrum slice $S_{04x}(f, v)$ given by (3.9) from the samples $r(k)$ for $k = 1, \dots, K_s$. From (3.9) we have

$$S_{04x}(f, v) = M_{04x}(f, v) - |S_{2x}(f, v)|^2 - |M_{2x}(f, v)|^2 - S_{d2x}(f)S_{d2x}(v) \quad (3.62)$$

where $M_{04x}(f, v)dfdv = E[|dX(f)dX(v)|^2]$, $S_{2x}(f, v)dfdv = E[dX(f)dX(v)^*]$, $M_{2x}(f, v)dfdv = E[dX(f)dX(v)]$ and $S_{d2x}(f)df = E[|dX(f)|^2]$. We see that estimation of $S_{04x}(f, v)$ requires estimation of $M_{04x}(f, v)$, $S_{2x}(f, v)$ and $M_{2x}(f, v)$. Since $S_{d2x}(f) = S_{2x}(f, f)$, we do not have to estimate $S_{d2x}(f)$ separately.

We first consider the estimation of $M_{04x}(f, v)$. Let us briefly explain its meaning. The slice $M_{04x}(f, v)$ is obtained from

$$M_{4x}(f_1, f_2, f_3)df_1df_2df_3 = E[dX(f_1)dX(f_2)dX(-f_3)^*dX(f_1 + f_2 + f_3)^*] \quad (3.63)$$

by setting $f_1 = f$, $f_2 = v$ and $f_3 = -v$. The spectrum $M_{4x}(f_1, f_2, f_3)$ is related to the fourth order statistic $R_{4x}(t, \tau_1, \tau_2, \tau_3) = E[x(t + \tau_1)x(t + \tau_2)x(t + \tau_3)^*x(t)^*]$. If $x(t)$ is

stationary, we have $R_{4x}(t, \tau_1, \tau_2, \tau_3) = R_{4x}(\tau_1, \tau_2, \tau_3)$ and then

$$M_{4x}(f_1, f_2, f_3) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} R_{4x}(\tau_1, \tau_2, \tau_3) e^{-j2\pi(f_1\tau_1 + f_2\tau_2 + f_3\tau_3)} d\tau_1 d\tau_2 d\tau_3. \quad (3.64)$$

If $x(t)$ is cyclostationary with the period T , we have

$$R_{4x}(t + T, \tau_1, \tau_2, \tau_3) = R_{4x}(t, \tau_1, \tau_2, \tau_3) = \sum_{q=-\infty}^{\infty} R_{4x}^{(q)}(\tau_1, \tau_2, \tau_3) e^{j2\pi \frac{q}{T} t} \quad (3.65)$$

where $R_{4x}^{(q)}(\tau_1, \tau_2, \tau_3) = \frac{1}{T} \int_0^T R_{4x}(t, \tau_1, \tau_2, \tau_3) e^{-j2\pi \frac{q}{T} t} dt$. In this case

$$M_{4x}(f_1, f_2, f_3) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} R_{4x}^{(0)}(\tau_1, \tau_2, \tau_3) e^{-j2\pi(f_1\tau_1 + f_2\tau_2 + f_3\tau_3)} d\tau_1 d\tau_2 d\tau_3. \quad (3.66)$$

These relations between $M_{4x}(f_1, f_2, f_3)$ and $R_{4x}(t, \tau_1, \tau_2, \tau_3)$ can be easily shown using the methodology from [70]. We estimate $M_{04r}(f, v)$ where the subscript r means that this spectrum is for the signal $r(k)$ for $k = 1, \dots, K_s$ as follows. We define the vectors $\mathbf{z}_n = \left[r(1 + (n-1)J_d) \quad \dots \quad r((n-1)J_d + J_{fb}) \right]^T$ for $n = 1, \dots, N_v$ where J_{fb} is size of the sliding window and $1 \leq J_d \leq J_{fb}$ is the parameter that determines how much the window is shifted in each step. Next, we compute the vectors $\mathbf{y}_n = FFT\{\mathbf{z}_n\}$ for $n = 1, \dots, N_v$ and the matrix

$$\hat{\mathbf{Y}}_{M4} = \frac{1}{N_v} \sum_{n=1}^{N_v} (\mathbf{y}_n \circ \mathbf{y}_n^*) (\mathbf{y}_n \circ \mathbf{y}_n^*)^T \quad (3.67)$$

whose entries are the estimates of $M_{04r}(f, v)$ over the J_{fb} by J_{fb} frequency grid. Hence, the size of the sliding window J_{fb} determines the number of frequency bins in spectrum estimation. We set $J_d = 1$ because then all vectors of the size J_{fb} from the input signal are used in the estimation procedure.

Next, we consider the estimation of $S_{2x}(f, v)$. But first we need to understand its meaning. The spectrum $S_{2x}(f, v)$ is related to the second order statistic $R_{2x}(t, \tau) = E[x(t + \tau)x(t)^*]$. If the signal $x(t)$ is stationary we have $R_{2x}(t, \tau) = R_{2x}(\tau)$. In this case the function $S_{2x}(f, f)$ is the Fourier transform of $R_{2x}(\tau)$ and it is the signal power spectrum. For $f \neq v$, $S_{2x}(f, v)$ is correlation between the spectral components at frequencies f and v . It is well known that for stationary signals $S_{2x}(f, v) = 0$ when $f \neq v$ [69]. If the signal $x(t)$ is cyclostationary with the period T then $R_{2x}(t+T, \tau) = R_{2x}(t, \tau)$. Hence, this statistic can be expanded as

$$R_{2x}(t, \tau) = \sum_{q=-\infty}^{\infty} R_{2x}^{(q)}(\tau) e^{j2\pi \frac{q}{T} t} \quad (3.68)$$

where $R_{2x}^{(q)}(\tau) = \frac{1}{T} \int_0^T R_{2x}(t, \tau) e^{-j2\pi \frac{q}{T} t} dt$. Let us define the spectra

$$S_{2x}^{(q)}(f) df = E[dX(f) dX(f - q/T)^*] \quad (3.69)$$

for $q = \dots, -2, -1, 0, 1, 2, \dots$. It is well known that

$$S_{2x}^{(q)}(f) = \int_{-\infty}^{\infty} R_{2x}^{(q)}(\tau) e^{-j2\pi f \tau} d\tau \quad (3.70)$$

for $q = \dots, -2, -1, 0, 1, 2, \dots$ [10]. The spectrum $S_{2x}^{(0)}(f)$ is the signal power spectrum and $S_{2x}^{(q)}(f)$ for $q \neq 0$ are the second order cyclostationary spectra. We see that $S_{2x}^{(q)}(f)$ are one dimensional slices of $S_{2x}(f, v)$ obtained for $f - v = q/T$ for $q = \dots, -2, -1, 0, 1, 2, \dots$. Therefore, $S_{2x}(f, v)$ contains power spectrum and it may contain second order cyclostationary spectra if any cyclostationary signals are present. Let us now make an important point about the second order cyclostationary spectra. We consider

$$\begin{aligned} R_{2x}^{(q)}(\tau)_{\Delta t} &= \frac{1}{T} \int_0^T R_{2x}(t + \Delta t, \tau) e^{-j2\pi \frac{q}{T} t} dt \\ &= \frac{e^{j2\pi \frac{q}{T} \Delta t}}{T} \int_{-\Delta t}^{T-\Delta t} R_{2x}(t, \tau) e^{-j2\pi \frac{q}{T} t} dt = e^{j2\pi \frac{q}{T} \Delta t} R_{2x}^{(q)}(\tau) \end{aligned} \quad (3.71)$$

where $0 \leq \Delta t \leq T$. It follows that also

$$S_{2x}^{(q)}(f)_{\Delta t} = \int_{-\infty}^{\infty} R_{2x}^{(q)}(\tau)_{\Delta t} e^{-j2\pi f \tau} d\tau = e^{j2\pi \frac{q}{T} \Delta t} \int_{-\infty}^{\infty} R_{2x}^{(q)}(\tau) e^{-j2\pi f \tau} d\tau = e^{j2\pi \frac{q}{T} \Delta t} S_{2x}^{(q)}(f). \quad (3.72)$$

We see that the spectra $S_{2x}^{(q)}(f)$ for $q \neq 0$ are sensitive to the timing parameter Δt . In other words, for $q \neq 0$ the spectra $S_{2x}^{(q)}(f)$ computed for different values of Δt differ even when the statistic $R_{2x}(t, \tau)$ is otherwise perfectly known. We also see that the power spectrum $S_{2x}^{(0)}(f)$ is not sensitive to Δt . According to (3.62) $S_{04x}(f, v)$ depends on $|S_{2x}(f, v)|^2$. Therefore, we need to correctly estimate magnitude of the second order cyclostationary spectra.

In our application the signal $r(k)$ for $k = 1, \dots, K_s$ consists of stationary noise and cyclostationary received signals. We assumed that the transmitted signals $s_m(t)$ for $m = 1, \dots, M$ are cyclostationary with periods $T_m = N_m T_s$ for $m = 1, \dots, M$. It follows that the signal $r(k)$ for $k = 1, \dots, K_s$ is cyclostationary with N_{com} , which is the least common

multiple of $\{N_1, \dots, N_M\}$. When estimating $S_{2r}(f, v)$ we have to take into account that it contains the second order cyclostationary spectra, which are sensitive to the timing of $r(k)$. We define the vectors $\mathbf{z}_n^{(p)} = \begin{bmatrix} r(p + (n-1)J_d) & \dots & r(p-1 + (n-1)J_d + J_{fb}) \end{bmatrix}^T$ for $n = 1, \dots, N_v^{(p)}$ where $p = 1, \dots, J_d$ and $J_d = N_{mul}N_{com}$ where $N_{mul} \geq 1$ is an integer. For each p we have the set of vectors $\mathbf{z}_n^{(p)}$ for $n = 1, \dots, N_v^{(p)}$ obtained by sliding the window with $J_d = N_{mul}N_{com}$, which assures that each vector in the set contains samples with the same timing of $r(k)$. We compute the vectors $\mathbf{y}_n^{(p)} = FFT\{\mathbf{z}_n^{(p)}\}$ for $n = 1, \dots, N_v^{(p)}$ and $p = 1, \dots, J_d$ and the matrices

$$\hat{\mathbf{Y}}_{S2}^{(p)} = \frac{1}{N_v^{(p)}} \sum_{n=1}^{N_v^{(p)}} \mathbf{y}_n^{(p)} \mathbf{y}_n^{(p)H}. \quad (3.73)$$

The entries of $\hat{\mathbf{Y}}_{S2}^{(p)}$ are the estimates of $S_{2r}(f, v)_p$ over the J_{fb} by J_{fb} frequency grid for one value of the discrete timing parameter p , which plays the same role as Δt in (3.71) and (3.72). The separate estimate $\hat{\mathbf{Y}}_{S2}^{(p)}$ for each $1 \leq p \leq J_d$ is needed because of the sensitivity of the cyclostationary spectra to the signal timing given by (3.71) and (3.72). If we computed one estimate by sliding the window with $J_d = 1$, as we did in the estimation of $M_{04r}(f, v)$, we would correctly estimate only the power spectrum slice $S_{2r}(f, f)$ while the estimates of $S_{2r}(f, v)$ for $f \neq v$ would be close to zero because of the dependencies in (3.71) and (3.72). In other words, we would completely miss all the cyclostationary spectra contained in $S_{2r}(f, v)$, which would then produce incorrect estimates of $S_{04r}(f, v)$ as can be seen from (3.62). The estimates of $M_{04r}(f, v)$ and the power spectrum $S_{2r}(f, f)$ computed by sliding the window with $J_d = 1$ are correct because these spectra are Fourier transforms of statistics that are not sensitive to the signal timing. From the diagonal entries of the matrices $\hat{\mathbf{Y}}_{S2}^{(p)}$ we compute the vector

$$\hat{\mathbf{y}}_{S2} = \begin{bmatrix} \frac{1}{J_d} \sum_{p=1}^{J_d} \hat{\mathbf{Y}}_{S2}^{(p)}(1, 1) \\ \vdots \\ \frac{1}{J_d} \sum_{p=1}^{J_d} \hat{\mathbf{Y}}_{S2}^{(p)}(J_{fo}, J_{fo}) \end{bmatrix} \quad (3.74)$$

whose entries are power spectrum estimates at J_{fb} frequency bins.

Finally, we consider the estimation of $M_{2x}(f, v)$. The spectrum $M_{2x}(f, v)$ is related to the second order statistic $R_{m2x}(t, \tau) = E[x(t + \tau)x(t)]$. If the signal $x(t)$ is real we have $R_{m2x}(t, \tau) = R_{2x}(t, \tau)$ and $M_{2x}(f, -v) = S_{2x}(f, v)$. Thus, for real signals these

two spectra contain the same information. In contrast, for complex signals $M_{2x}(f, v)$ and $S_{2x}(f, v)$ contain different information. If the signal $x(t)$ is stationary we have $R_{m2x}(t, \tau) = R_{m2x}(\tau)$. In this case the function $M_{2x}(f, -f)$ is the Fourier transform of $R_{m2x}(\tau)$ but this is not the signal power spectrum. For stationary signals we have $M_{2x}(f, v) = 0$ when $f + v \neq 0$. If the signal $x(t)$ is cyclostationary with the period T then $R_{m2x}(t + T, \tau) = R_{m2x}(t, \tau)$. Hence, this statistic can be expanded as

$$R_{m2x}(t, \tau) = \sum_{q=-\infty}^{\infty} R_{m2x}^{(q)}(\tau) e^{j2\pi \frac{q}{T} t} \quad (3.75)$$

where $R_{m2x}^{(q)}(\tau) = \frac{1}{T} \int_0^T R_{m2x}(t, \tau) e^{-j2\pi \frac{q}{T} t} dt$. Let us define the spectra

$$M_{2x}^{(q)}(f) df = E[dX(f) dX(-f + q/T)] \quad (3.76)$$

for $q = \dots, -2, -1, 0, 1, 2, \dots$. In this case we have

$$M_{2x}^{(q)}(f) = \int_{-\infty}^{\infty} R_{m2x}^{(q)}(\tau) e^{-j2\pi f \tau} d\tau \quad (3.77)$$

for $q = \dots, -2, -1, 0, 1, 2, \dots$. These relations can be shown using the same methodology that was used for $S_{2x}(f, v)$ in [10]. We see that $M_{2x}^{(q)}(f)$ are one dimensional slices of $M_{2x}(f, v)$ obtained for $f + v = q/T$ for $q = \dots, -2, -1, 0, 1, 2, \dots$. Again, $M_{2x}^{(0)}(f)$ is not sensitive to the signal timing, whereas $M_{2x}^{(q)}(f)$ for $q \neq 0$ are sensitive to the signal timing in a way similar to (3.71) and (3.72). Hence, our estimation procedure for $M_{2r}(f, v)$ will be similar to the one used for $S_{2r}(f, v)$. We compute the matrices

$$\hat{\mathbf{Y}}_{M2}^{(p)} = \frac{1}{N_v^{(p)}} \sum_{n=1}^{N_v} \mathbf{y}_n^{(p)} \mathbf{y}_n^{(p)T} \quad (3.78)$$

for $p = 1, \dots, J_d$. The entries of $\hat{\mathbf{Y}}_{M2}^{(p)}$ are the estimates of $M_{2r}(f, v)_p$ over the J_{fb} by J_{fb} frequency grid for one value of the discrete timing parameter p , which plays the same role as in (3.73). Now we can compute the desired estimate of $S_{04x}(f, v)$. We compute the matrix

$$\hat{\mathbf{Y}}_{0S4} = \hat{\mathbf{Y}}_{0M4} - \frac{1}{J_d} \sum_{p=1}^{J_d} \left[\hat{\mathbf{Y}}_{S2}^{(p)} \circ \hat{\mathbf{Y}}_{S2}^{(p)*} \right] - \frac{1}{J_d} \sum_{p=1}^{J_d} \left[\hat{\mathbf{Y}}_{M2}^{(p)} \circ \hat{\mathbf{Y}}_{M2}^{(p)*} \right] - \hat{\mathbf{y}}_{S2} \hat{\mathbf{y}}_{S2}^T. \quad (3.79)$$

whose entries are the estimates of $S_{04r}(f, v)$ over the J_{fb} by J_{fb} frequency grid.

We illustrate the proposed estimation procedure with two examples. In the first example the signal $r(k)$ is complex white circularly symmetric Gaussian noise. This signal has real valued Gaussian processes as its real and imaginary part and has the following properties $R_{2r}(k, m) = E[r(k+m)r(k)^*] = \sigma^2 \delta_{km}$ where σ^2 is the signal variance and δ_{km} is the Kronecker delta function and $R_{m2r}(k, m) = E[r(k+m)r(k)] = 0$ for all k and m . This implies that real and imaginary parts have the same statistics and they are mutually uncorrelated and hence statistically independent [87, sec. 3.2.8]. Figure 3.1 shows the estimate of $|S_{2r}(f, v)|$. This signal is stationary and hence, we should have $S_{2r}(f, v) = 0$ for $f \neq v$. The one dimensional slice $S_{2r}(f, f)$ is the signal power spectrum, which should be flat in frequency because the signal is white noise. Figure 3.2 shows the estimate of $|M_{2r}(f, v)|$. For this signal we have $R_{m2r}(k, m) = E[r(k+m)r(k)] = 0$ for all k and m and hence, $M_{2r}(f, v) = 0$ for all f and v . Figure 3.3 and 3.4 show the estimates of $M_{04r}(f, v)$ and $S_{04r}(f, v)$. Since the signal is Gaussian its fourth order cumulant spectrum is zero and hence, $S_{04r}(f, v) = 0$ for all f and v . We see that the estimates shown in Figures 3.1, 3.2, 3.3 and 3.4 agree with the theory. We also see that $S_{2r}(f, v)$ and $M_{04r}(f, v)$ consist of discrete component along the line $f = v$ and a smooth background function. Thus, they are not smooth functions of f and v . On the other hand, $S_{2r}(f, v)$ is a smooth function of f and v .

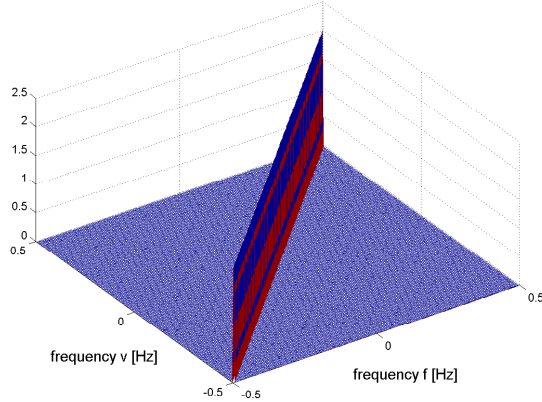


Figure 3.1: Estimate of the second order spectrum $|S_{2r}(f, v)|$ for complex Gaussian noise

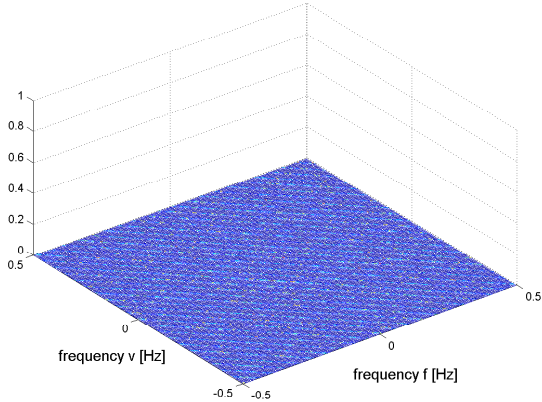


Figure 3.2: Estimate of the second order spectrum $|M_{2r}(f, v)|$ for complex Gaussian noise

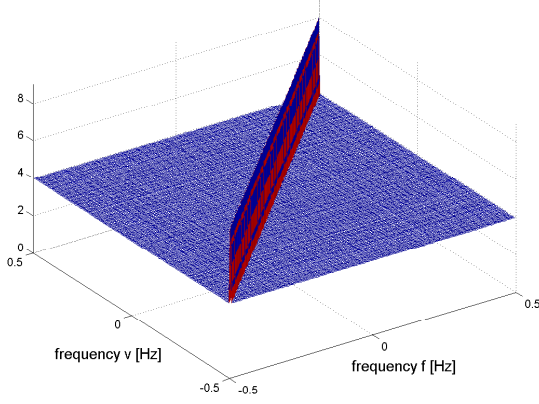


Figure 3.3: Estimate of the fourth order moment spectrum slice $M_{04r}(f, v)$ for complex Gaussian noise

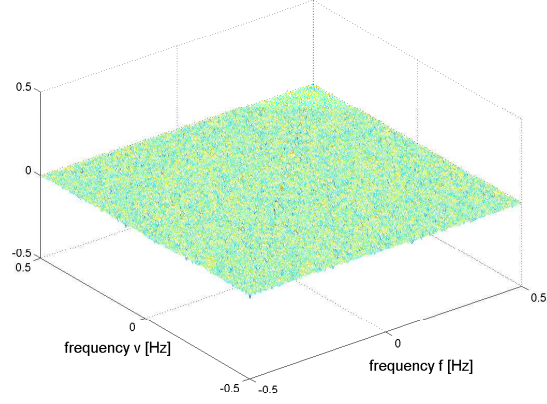


Figure 3.4: Estimate of the fourth order cumulant spectrum slice $S_{04r}(f, v)$ for complex Gaussian noise

In our second example we consider the DBPSK signal with Barker sequence spreading, which is used in 802.11b networks [67]. This signal is of the form (3.13) with the symbol period $T = 1\mu s$. The discrete time signal $r(k)$ is obtained by sampling the continuous time signal with $T_s = 0.05\mu s$. The estimation results for this signal are shown in Figures 3.5, 3.6, 3.7 and 3.8. This signal is cyclostationary with the period T . The spectrum $S_{2r}(f, v)$ has nonzero one dimensional slices at $f - v = q/T$ for $q = \dots, -2, -1, 0, 1, 2, \dots$ and it is zero otherwise. The slice obtained for $q = 0$ is the signal power spectrum and the slices obtained for $q \neq 0$ are the second order cyclostationary spectra. The spectrum $M_{2r}(f, v)$ has nonzero one dimensional slices at $f + v = q/T$ for $q = \dots, -2, -1, 0, 1, 2, \dots$ and it is zero otherwise. Since the signal $r(k)$ is complex the one dimensional slices of $M_{2r}(f, v)$ are different from those of $S_{2r}(f, v)$. Now the signal $r(k)$ is not Gaussian and $S_{04r}(f, v) \neq 0$. We see that $S_{2r}(f, v)$, $M_{2r}(f, v)$, and $M_{04r}(f, v)$ are not smooth functions of f and v . The moment spectrum slice $M_{04r}(f, v)$ consists of a smooth background function and discrete one dimensional components along the lines $f - v = q/T$ and $f + v = q/T$ for $q = \dots, -2, -1, 0, 1, 2, \dots$. In contrast to this, the cumulant spectrum slice $S_{04r}(f, v)$ is a smooth function of f and v . In general, moment spectra contain discrete components and are not smooth

functions of frequency. When cumulant spectra are computed these discrete components are subtracted and the resulting spectrum is smooth. More details on this can be found in [37].

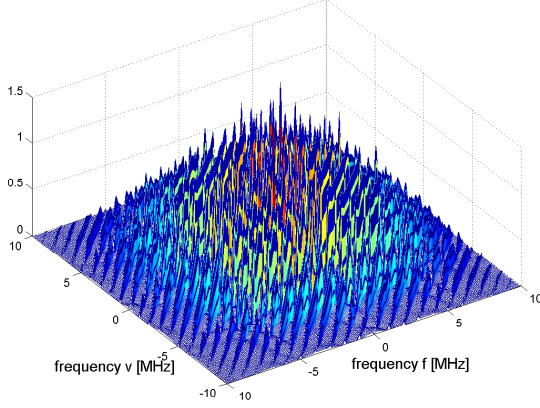


Figure 3.5: Estimate of the second order spectrum $|S_{2r}(f, v)|$ for DBPSK signal with Barker sequence spreading

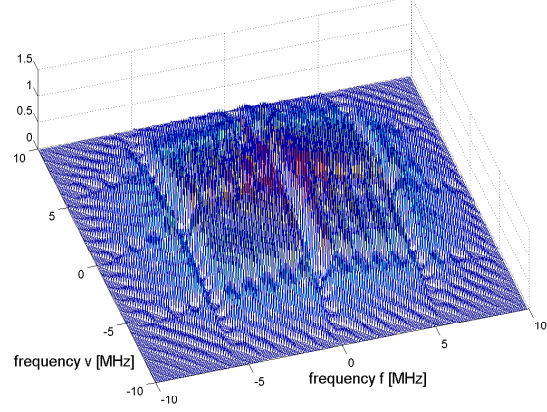


Figure 3.6: Estimate of the second order spectrum $|M_{2r}(f, v)|$ for DBPSK signal with Barker sequence spreading

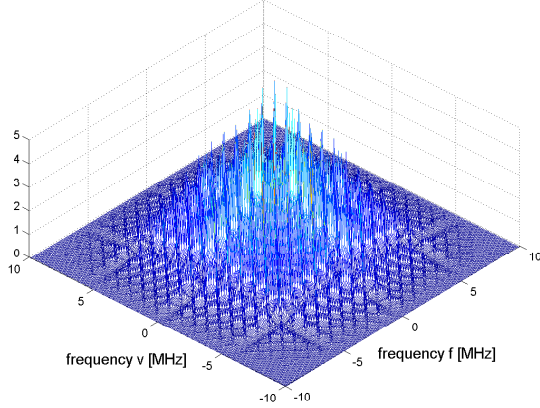


Figure 3.7: Estimate of the fourth order moment spectrum slice $M_{04r}(f, v)$ for DBPSK signal with Barker sequence spreading

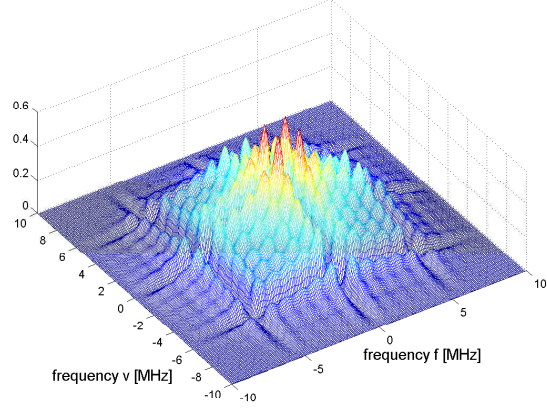


Figure 3.8: Estimate of the fourth order cumulant spectrum slice $|S_{04r}(f, v)|$ for DBPSK signal with Barker sequence spreading

Our estimation procedure uses the method of time averaging. We slide a window over the received signal and for each window position in time we compute the FFT coefficients and approximate the computed statistic by replacing the spectral process

in the definition with the corresponding FFT coefficients. Final estimate is obtained by computing the average value over all window positions. Time averaging methods for power spectrum estimation are well known and widely used [34]. Main difference here is that some statistics of interest are sensitive to signal timing because of the presence of cyclostationary signals. When these statistics are computed, the time averaging method has to be modified as we explained here. Otherwise, we obtain completely incorrect estimates.

The second class spectrum estimation methods is based on frequency averaging (smoothing), where the initial estimate is smoothed using various algorithms. Typically, estimate at any given frequency is computed as an average value over several neighboring frequencies. The estimates described here are obtained without any frequency smoothing. In some cases frequency smoothing may be needed to reduce variance of the estimate. Frequency smoothing can be added into our estimation procedure as follows. First approach is to apply frequency smoothing on the estimates of $M_{04r}(f, v)$, $M_{2r}(f, v)$ and $S_{2r}(f, v)$ obtained using the time averaging method and use these smoothed estimates to estimate $S_{04r}(f, v)$. When applying frequency smoothing on the estimates of $M_{04r}(f, v)$, $M_{2r}(f, v)$ and $S_{2r}(f, v)$ we have to be careful because these spectra are not smooth themselves and often contain various discrete components. For example, when smoothing the estimate of $S_{2r}(f, v)$ shown in Figure 3.5 we have to preserve the discrete one dimensional slices at $f - v = q/T$ for $q = \dots, -2, -1, 0, 1, 2, \dots$. Here we cannot use simple linear methods for smoothing such as those proposed in [85] or [86], because they would smear any discrete components present in this spectrum. The nonzero one dimensional components present in $S_{2r}(f, v)$ can be smoothed along the lines $f - v = q/T$ for $q = \dots, -2, -1, 0, 1, 2, \dots$, but not in any other direction. Similarly, the nonzero one dimensional components present in $M_{2r}(f, v)$ whose estimate is shown in Figure 3.6 can be smoothed along the lines $f + v = q/T$ for $q = \dots, -2, -1, 0, 1, 2, \dots$, but not in any other direction. Applying smoothing on the estimate of $M_{04r}(f, v)$ shown in Figure 3.6 is even more complicated because here we have discrete components along the lines $f - v = q/T$ and $f + v = q/T$ for $q = \dots, -2, -1, 0, 1, 2, \dots$. Therefore, applying

smoothing on the estimates of $M_{04r}(f, v)$, $M_{2r}(f, v)$ and $S_{2r}(f, v)$ is complicated because of the discrete components present in these spectra. This problem with applying linear smoothers on the spectra with discrete components was also pointed out in [38], where the authors propose some modifications of linear smoothing methods in order to preserve discrete components present in the estimated spectra. The alternative way to add frequency smoothing into our procedure is to use the estimates of $M_{04r}(f, v)$, $M_{2r}(f, v)$ and $S_{2r}(f, v)$ obtained using the time averaging method to find an initial estimate of $S_{04r}(f, v)$ and then apply frequency smoothing on this estimate. Since the spectrum $S_{04r}(f, v)$ is in most cases a smooth function of f and v [37], we can use the following linear smoothing method

$$\hat{\mathbf{Y}}_{0S4}^{(fs)}(i, j) = \frac{1}{(2L_{sm} + 1)^2} \sum_{p=i-L_{sm}}^{i+L_{sm}} \sum_{q=i-L_{sm}}^{i+L_{sm}} \hat{\mathbf{Y}}_{0S4}(p, q) \quad (3.80)$$

where $i = 1, \dots, J_{fb}$, $j = 1, \dots, J_{fb}$, $\hat{\mathbf{Y}}_{0S4}$ is our initial estimate given by (3.79), $\hat{\mathbf{Y}}_{0S4}^{(fs)}$ is the smoothed estimate and L_{sm} is the parameter controlling how much frequency smoothing is applied.

Another approach for fourth order spectrum estimation is to use the fact that $S_{4r}(f_1, f_2, f_3)$ defined in (3.3) is the Fourier transform of $C_{4r}(m, l, k) = cum(r(n + m), r(n + l), r(n + k)^*, r(n)^*)$. Using $r(k)$ for $k = 1, \dots, K_s$ we need to estimate $C_{4r}(m, l, k)$ for $-L_{lag} \leq m, l, k \leq L_{lag}$ where $J_{fb} = 2L_{lag} + 1$. We apply three dimensional FFT on $C_{4r}(m, l, k)$ to obtain $S_{4r}(f_1, f_2, f_3)$ and then we find $S_{04r}(f, v) = S_{4r}(f, v, -v)$. This approach requires more computations since we have to compute the whole fourth order spectrum first and then extract the desired slice. When estimating $C_{4r}(m, l, k)$ we cannot use the procedure for stationary signals outlined in [36, ch. 4] because it ignores any cyclostationary spectra present in $M_{2r}(f, v)$ and $S_{2r}(f, v)$ and thus, produces incorrect estimates. Therefore, this procedure also has to be modified to accommodate cyclostationary signals.

So far we have considered the problem of fourth order spectrum estimation using the samples of $r(k)$ from one continuous time segment. Let us now consider the same problem where we have samples from two or more disjoint time segments. Let the l -th continuous segment contain the samples $r(k)$ for $K_l < k \leq K_l + K_{s,l}$ where $l = 1, \dots, L_{cs}$

and $1 \leq K_1 < K_1 + K_{s,1} < K_2 < K_2 + K_{s,2} < \dots < K_{L_{cs}} < K_{L_{cs}} + K_{s,L_{cs}} \leq K$. We first assume for simplicity that these samples were generated by one transmitter transmitting L_{cs} packets during the observation interval. We assume that the transmitted signals during different time segments are statistically identical except for the possible variation in timing of cyclostationary statistics across the segments. In other words, the timing parameter Δt that appears in (3.71) may be different for each segment. We know that $M_{04r}(f, v)$ and $S_{2r}^{(0)}(f)$ are not sensitive to the signal timing. We also know that changes in Δt only affect phases of the entries of $S_{2r}(f, v)$ and $M_{2r}(f, v)$ according to (3.72). Since $S_{04r}(f, v) = M_{04r}(f, v) - |S_{2r}(f, v)|^2 - |M_{2r}(f, v)|^2 - S_{2r}^{(0)}(f)S_{2r}^{(0)}(v)$, it follows that $S_{04r}(f, v)$ is not sensitive to the signal timing. This analysis assumed one cyclostationary signal. Let us now assume the above defined segments are generated as a sum of M_{sg} transmitted signals where each segment has the same combination of active transmitted signals. Each transmitter transmits the same signal during all the segments, but the timing of the cyclostationary statistics may vary across the segments. We further assume that the transmitted signals are statistically independent on each of the segments. Since $S_{04r}(f, v)$ is one slice of the fourth order cumulant spectrum $S_{4r}(f_1, f_2, f_3)$ and the signals are statistically independent we have for each segment $S_{04r}(f, v) = \sum_{m=1}^{M_{sg}} S_{04m}(f, v)$ where $S_{04m}(f, v)$ is the spectrum slice of the m -th individual signal. Since we already know that each $S_{04m}(f, v)$ for $m = 1, \dots, M_{sg}$ is insensitive to the timing of cyclostationary statistics it follows that $S_{04r}(f, v)$ is also insensitive to this effect. Therefore, if the above defined segments of $r(k)$ are generated by a same combination of active transmitted signals we can use the samples from different segments to estimate $S_{04r}(f, v)$ even if the timing parameters of cyclostationary statistics of individual signals vary across the segments. The spectrum $S_{04r}(f, v)$ can be estimated from the above defined segments containing the samples of $r(k)$ for $K_l < k \leq K_l + K_{s,l}$ where $l = 1, \dots, L_{cs}$ as follows. We compute the estimates $\hat{\mathbf{Y}}_{0S4,l}$ using (3.79) for each of the segments where $l = 1, \dots, L_{cs}$. Then we find the estimate for all the segments as

$$\hat{\mathbf{Y}}_{0S4} = \sum_{l=1}^{L_{cs}} \omega_l \hat{\mathbf{Y}}_{0S4,l} \quad (3.81)$$

where $\omega_l = K_{s,1} / \sum_{q=1}^{L_{cs}} K_{s,q}$. The final estimate is obtained by applying frequency smoothing defined by (3.80) on $\hat{\mathbf{Y}}_{0S4}$ from (3.81). This estimate is computed for the set of segments containing the samples of $r(k)$ generated by the same combination of active transmitted signals. Let us assume that there are I_{cl} such sets of segments in the signal $r(k)$ for $k = 1, \dots, K$ where each set of segments corresponds to one fixed combination of transmitted signals. We apply the described estimation procedure on each set of the segments of $r(k)$ and denote the resulting estimates $\hat{\mathbf{Y}}_i$ for $i = 1, \dots, I_{cl}$. These matrices are arranged in the J_{fb} by J_{fb} by I_{cl} three way array $\hat{\mathbf{Y}}$, which is an estimate of \mathbf{Y} defined in (3.11). The estimate $\hat{\mathbf{Y}}$ is needed for application of the signal analysis method proposed in the previous section.

Our estimation procedure assumes that the possible set of values of the periods T_m for $m = 1, \dots, M$ associated with the cyclostationary transmitted signals is known. This knowledge is needed for selection of the sampling period T_s which is used to obtain the discrete time signal $r(k)$ for $k = 1, \dots, K$ from the continuous time signal $r(t)$ from (2.1). The sampling period must satisfy $T_m = N_m T_s$ where T_m is the period associated with the transmitted signal $s_m(t)$ from (2.1) and N_m is an integer such that $N_m \geq 2$ for $m = 1, \dots, M$. If this condition is not satisfied the statistics of $r(k)$ may be completely different from those of $r(t)$ as explained earlier. Obviously, in this case the proposed analysis is not meaningful. The above condition on the sampling period is implicitly assumed in any type of statistical analysis of cyclostationary signals. The knowledge of the periods T_m is also needed for selection of the parameter J_d that determines the shifts of the sliding window in estimation of $S_{2r}(f, v)$ and $M_{2r}(f, v)$. As explained earlier $J_d = N_{mul} N_{com}$ where N_{com} is the least common multiple of $\{N_1, \dots, N_M\}$ and $N_{mul} \geq 1$ is an integer. If we have no knowledge on the possible values of the periods T_m it is not possible to select appropriate values for T_s and J_d and the proposed estimation procedure cannot be applied. This estimation procedure also requires selection of the number of frequency bins J_{fb} and the smoothing factor L_{sm} from (3.80). Increasing J_{fb} improves frequency resolution, but it also increases variance of the obtained estimate. Increasing L_{sm} reduces the variance, but it also introduces bias in the resulting estimate.

3.3 Computational methods

Using the procedure from the previous section we can compute the estimate $\hat{\mathbf{Y}}$. In this section we develop numerical algorithms for computing estimates of the matrices \mathbf{F} , \mathbf{C} and \mathbf{K} from $\hat{\mathbf{Y}}$. These algorithms are needed for practical application of the fourth order spectrum based analysis method proposed earlier. There are various algorithm for fitting of three way arrays to different types of models [84] [76] [88] [89] [90]. Since our problem has some unique aspects, none of these algorithms can be directly applied here.

We start with the following weighted least squares criterion

$$C_{wls} = \sum_{j=1}^{J_{fb}} \sum_{n=j}^{J_{fb}} \sum_{i=1}^{I_{cl}} \gamma_i \left(\hat{\mathbf{Y}}(j, n, i) - \sum_{m=1}^M \sum_{r=1}^R k_{rm} f_{jr} f_{nr} c_{im} \right)^2 \quad (3.82)$$

where $\hat{\mathbf{Y}}(j, n, i)$ is the observed (estimated) value given by $\hat{\mathbf{Y}}(j, n, i) = \mathbf{Y}(j, n, i) + \mathbf{E}(j, n, i)$ where $\mathbf{E}(j, n, i)$ is the estimation error. In (3.82) we also used the symmetry $\hat{\mathbf{Y}}(j, n, i) = \hat{\mathbf{Y}}(n, j, i)$. One reasonable choice for weights is $\gamma_i = \sum_{k=1}^K \hat{\mathbf{S}}(i, k)/K$ where $\hat{\mathbf{S}}$ is the estimated matrix of the segmentation sequences. This choice makes sense because the matrix $\hat{\mathbf{Y}}_i$ is estimated from the samples of $r(k)$ located at time instants where the i -th row of $\hat{\mathbf{S}}$ has ones. We seek the parameters $\mathbf{F} = [f_{jr}]$, $\mathbf{C} = [c_{im}]$, and $\mathbf{K} = [k_{rm}]$ that minimize C_{wls} subject to the constraints $f_{jr} \geq 0$ and $c_{im} \geq 0$. In other words, we seek the estimates satisfying

$$[\hat{\mathbf{F}}, \hat{\mathbf{C}}, \hat{\mathbf{K}}] = \underset{\mathbf{F} \geq 0, \mathbf{C} \geq 0, \mathbf{K}}{\operatorname{argmin}} C_{wls}(\mathbf{F}, \mathbf{C}, \mathbf{K}). \quad (3.83)$$

If we multiply the r -th column of \mathbf{F} with some constant $\alpha \neq 0$ and divide the r -th row of \mathbf{K} with α^2 the entries of \mathbf{Y} from (3.37) do not change. Also, if we multiply the m -th column of \mathbf{C} with some constant $\alpha \neq 0$ and divide the m -th column of \mathbf{K} with α the entries of \mathbf{Y} from (3.37) do not change. These facts imply that the optimization problem (3.83) does not have a unique solution unless we impose additional constraints on the estimated parameters. The simplest possibility is to impose unit l_1 norm constraints on the columns of \mathbf{F} and \mathbf{C} . Thus, we seek the estimates

$$[\hat{\mathbf{F}}, \hat{\mathbf{C}}, \hat{\mathbf{K}}] = \underset{\mathbf{F} \geq 0, \mathbf{C} \geq 0, \mathbf{K}, \mathbf{1}^T \mathbf{F} = \mathbf{1}^T, \mathbf{1}^T \mathbf{C} = \mathbf{1}^T}{\operatorname{argmin}} C_{wls}(\mathbf{F}, \mathbf{C}, \mathbf{K}) \quad (3.84)$$

where $\mathbf{1}$ denotes the vector of all ones of the appropriate size.

In order to minimize the function $C_{wls}(\mathbf{F}, \mathbf{C}, \mathbf{K})$ defined in (3.82) we need to know the parameters M and R . In theory, these parameters can be determined from the ranks of the matrices defined in (3.53) and (3.58). If \mathbf{Y} is perfectly known M and R can be found from SVDs of the matrices in (3.53) and (3.58). In any practical situation only an estimate $\hat{\mathbf{Y}}$, containing some amount of error, is available. In that case M and R can be found as effective ranks of the matrices given by (3.53) and (3.58) where \mathbf{Y} is replaced with $\hat{\mathbf{Y}}$. These problems are very difficult themselves and require special algorithms such as information theoretic criteria for signal detection [91]. We shall discuss this problem further in the next section. In this section the parameters M and R are assumed to be known.

The optimization problem defined in (3.84) may not have a unique solution. Out of all possible solutions we seek the one where the matrix \mathbf{K} has the largest possible number of zero entries. Hence, we seek the estimates

$$[\hat{\mathbf{F}}, \hat{\mathbf{C}}, \hat{\mathbf{K}}] = \underset{\mathbf{F} \geq 0, \mathbf{C} \geq 0, \mathbf{K}, \mathbf{1}^T \mathbf{F} = \mathbf{1}^T, \mathbf{1}^T \mathbf{C} = \mathbf{1}^T, \min \|\mathbf{K}\|_0}{\operatorname{argmin}} C_{wls}(\mathbf{F}, \mathbf{C}, \mathbf{K}) \quad (3.85)$$

In other words, we select the solution where the matrix \mathbf{K} is maximally sparse. Since it is very difficult to find this solution directly, instead of (3.85) we consider the following optimization problem

$$[\hat{\mathbf{F}}, \hat{\mathbf{C}}, \hat{\mathbf{K}}] = \underset{\mathbf{F} \geq 0, \mathbf{C} \geq 0, \mathbf{K}, \mathbf{1}^T \mathbf{F} = \mathbf{1}^T, \mathbf{1}^T \mathbf{C} = \mathbf{1}^T}{\operatorname{argmin}} [C_{wls}(\mathbf{F}, \mathbf{C}, \mathbf{K}) + \beta \|\mathbf{K}\|_1] \quad (3.86)$$

where the parameter β controls the influence of the added regularization term. The solution of (3.86) is a compromise between minimizing $C_{wls}(\mathbf{F}, \mathbf{C}, \mathbf{K})$ and keeping $\|\mathbf{K}\|_1$ as small as possible. Intuitively, l_1 norm is nearest to l_0 norm and hence, we expect that the solution with the minimal $\|\mathbf{K}\|_1$ should be close to the one with the minimal $\|\mathbf{K}\|_0$. The advantage of considering (3.86) instead of (3.85) is that the former problem is much simpler to solve. In other words, the problem (3.86) is an analytically tractable relaxation of the problem (3.85). This technique of substituting l_0 norm with l_1 norm is widely used for enforcing sparsity constraints [92] [93]. The matrix \mathbf{K} can be represented as

$$\mathbf{K} = \mathbf{K}_p - \mathbf{K}_n \quad (3.87)$$

where the matrices \mathbf{K}_p and \mathbf{K}_n have only nonnegative entries. Using (3.87) we get

$$[\hat{\mathbf{F}}, \hat{\mathbf{C}}, \hat{\mathbf{K}}_p, \hat{\mathbf{K}}_n] = \quad (3.88)$$

$$\underset{\mathbf{F} \geq 0, \mathbf{C} \geq 0, \mathbf{K}_p \geq 0, \mathbf{K}_n \geq 0, \mathbf{1}^T \mathbf{F} = \mathbf{1}^T, \mathbf{1}^T \mathbf{C} = \mathbf{1}^T}{\operatorname{argmin}} [C_{wls}(\mathbf{F}, \mathbf{C}, \mathbf{K}_p, \mathbf{K}_n) + \beta(\|\mathbf{K}_p\|_1 + \|\mathbf{K}_n\|_1)].$$

The technique used for transforming (3.86) into (3.88) is well known [33] [94].

Our first step in solving the optimization problem (3.88) is to take into account the nonnegativity constraints using logarithmic barrier function, which is a well known technique for transforming a problem with inequality constraints into an unconstrained problem. A very good overview of this technique can be found in [56, ch. 11]. We define the parameter vector

$$\mathbf{p} = \begin{bmatrix} p_1 & \dots & p_{N_{par}} \end{bmatrix}^T = \begin{bmatrix} \operatorname{vec}(\mathbf{F})^T & \operatorname{vec}(\mathbf{C})^T & \operatorname{vec}(\mathbf{K}_p)^T & \operatorname{vec}(\mathbf{K}_n)^T \end{bmatrix}^T \quad (3.89)$$

and express the equality constraints $\mathbf{1}^T \mathbf{F} = \mathbf{1}^T$ and $\mathbf{1}^T \mathbf{C} = \mathbf{1}^T$ as $\mathbf{A}_{eq} \mathbf{p} = \mathbf{1}$ where \mathbf{A}_{eq} is a $2M$ by N_{par} matrix of zeros and ones where $N_{par} = J_{fb}R + I_{cl}M + 2RM$ is the total number of unknown parameters. We consider the new criterion function

$$C_{bf}(\mathbf{p}) = C_{wls}(\mathbf{p}) - \alpha \sum_{r=1}^{N_{par}} \log p_r + \beta \sum_{r=J_{fb}R+I_{cl}M+1}^{N_{par}} p_r \quad (3.90)$$

where α is a positive constant, which regulates influence of the added logarithmic terms. When any of the parameters with the nonnegativity constraint, say p_r , approaches zero its logarithmic term tends to negative infinity. Hence, C_{bf} tends to plus infinity, which prevents that parameter from becoming zero or negative. Our optimization problem now becomes

$$\hat{\mathbf{p}} = \underset{\mathbf{A}_{eq} \mathbf{p} = \mathbf{1}}{\operatorname{argmin}} C_{bf}(\mathbf{p}). \quad (3.91)$$

The parameter α should be chosen so that $C_{wls}(\hat{\mathbf{p}}) \gg C_{log}(\hat{\mathbf{p}})$ and $C_{reg}(\hat{\mathbf{p}}) \gg C_{log}(\hat{\mathbf{p}})$ where $C_{log}(\mathbf{p}) = -\alpha \sum_{r=1}^{N_{par}} \log p_r$ and $C_{reg}(\mathbf{p}) = \beta \sum_{r=J_{fb}R+I_{cl}M+1}^{N_{par}} p_r$. in other words, the logarithmic terms should be used for enforcing the nonnegativity constraints without affecting the computed solution significantly.

Let us try to solve (3.91) using Newton's method. Starting from an initial point we update parameter vector \mathbf{p} in each step as follows. Let \mathbf{p}_0 be the current parameter

vector. We approximate the objective function around the current point using quadratic approximation

$$C_{bf}(\mathbf{p}_0 + \Delta\mathbf{p}) \approx C_{bf}(\mathbf{p}_0) + \mathbf{g}(\mathbf{p}_0)^T \Delta\mathbf{p} + \frac{1}{2} \Delta\mathbf{p}^T \mathbf{H}(\mathbf{p}_0) \Delta\mathbf{p} \quad (3.92)$$

where $\mathbf{g}(\mathbf{p})$ is the gradient and $\mathbf{H}(\mathbf{p})$ is the Hessian of $C_{bf}(\mathbf{p})$. The updated parameter vector is

$$\mathbf{p} = \mathbf{p}_0 + \eta \Delta\mathbf{p} \quad (3.93)$$

where the step $\Delta\mathbf{p}$ minimizes the quadratic approximation (3.92) under the constraint $\mathbf{A}_{eq}\mathbf{p} = \mathbf{1}$ and η is the step size. Since the equality constraint must be satisfied in every iteration, we have $\mathbf{A}_{eq}\mathbf{p} = \mathbf{A}_{eq}(\mathbf{p}_0 + \Delta\mathbf{p}) = \mathbf{A}_{eq}\mathbf{p}_0 = \mathbf{1}$, which implies $\mathbf{A}_{eq}\Delta\mathbf{p} = \mathbf{0}$. Thus, we need to solve

$$\min_{\Delta\mathbf{p}, \mathbf{A}_{eq}\Delta\mathbf{p}=\mathbf{0}} \mathbf{g}(\mathbf{p}_0)^T \Delta\mathbf{p} + \frac{1}{2} \Delta\mathbf{p}^T \mathbf{H}(\mathbf{p}_0) \Delta\mathbf{p} \quad (3.94)$$

Solution of the problem (3.94) is well known [56, pp. 526]. We form the Lagrangian $L(\Delta\mathbf{p}, \boldsymbol{\lambda}) = \mathbf{g}(\mathbf{p}_0)^T \Delta\mathbf{p} + \frac{1}{2} \Delta\mathbf{p}^T \mathbf{H}(\mathbf{p}_0) \Delta\mathbf{p} + \boldsymbol{\lambda}^T \mathbf{A}_{eq}\Delta\mathbf{p}$ where $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers, compute its gradient and set the gradient to zero. We obtain

$$\begin{bmatrix} \mathbf{H}(\mathbf{p}_0) & \mathbf{A}_{eq}^T \\ \mathbf{A}_{eq} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{p} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\mathbf{g}(\mathbf{p}_0) \\ \mathbf{0} \end{bmatrix} \quad (3.95)$$

where the second equation takes into account the equality constraints. The optimal step $\Delta\mathbf{p}$ is found by solving (3.95). This procedure is equivalent to elimination of equality constraints and performing the minimization in (3.94) over $\Delta\mathbf{p} \in \text{Null}(\mathbf{A}_{eq})$. The solution for $\Delta\mathbf{p}$ computed from (3.95) is valid if $\Delta\mathbf{p}^T \mathbf{H}(\mathbf{p}_0) \Delta\mathbf{p} > 0$ for any $\Delta\mathbf{p} \in \text{Null}(\mathbf{A}_{eq})$. More detailed analysis of this method can be found in [56, Ch. 10].

Next, we need to compute the gradient $\mathbf{g}(\mathbf{p})$ and the Hessian $\mathbf{H}(\mathbf{p})$ of the criterion function $C_{bf}(\mathbf{p})$ given by (3.84). Let us define the vectors

$$\begin{aligned} \hat{\mathbf{y}} &= \begin{bmatrix} \text{vecr}(\hat{\mathbf{Y}}_1)^T & \dots & \text{vecr}(\hat{\mathbf{Y}}_{I_{cl}})^T \end{bmatrix}^T \\ \mathbf{y}(\mathbf{p}) &= \begin{bmatrix} \text{vecr}(\mathbf{Y}_1)^T & \dots & \text{vecr}(\mathbf{Y}_{I_{cl}})^T \end{bmatrix}^T \end{aligned} \quad (3.96)$$

where we emphasized the functional dependence of the model vector $\mathbf{y}(\mathbf{p})$ on the parameter vector. The functional dependence is given by (3.36) and (3.89). Now we

have

$$C_{bf}(\mathbf{p}) = \sum_{q=1}^{N_{eq}} w_q [\hat{y}_q - y_q(\mathbf{p})]^2 - \alpha \sum_{r=1}^{N_{par}} \log p_r + \beta \sum_{r=J_{fb}R+I_{cl}M+1}^{N_{par}} p_r \quad (3.97)$$

where $N_{eq} = I_{cl}(J_{fb} + \binom{J_{fb}}{2})$ is the total number of observations, \hat{y}_q and $y_q(\mathbf{p})$ are q -th entries of $\hat{\mathbf{y}}$ and $\mathbf{y}(\mathbf{p})$ respectively, p_r is the r -th entry of \mathbf{p} and w_q is the q -th diagonal entry of the matrix

$$\mathbf{\Gamma} = \text{diag} \left(\begin{bmatrix} \gamma_1 & \dots & \gamma_1 & \dots & \gamma_{I_{cl}} & \dots & \gamma_{I_{cl}} \end{bmatrix} \right). \quad (3.98)$$

Using (3.84) and (3.97) we find the gradient of $C_{bf}(\mathbf{p})$ as

$$\mathbf{g}(\mathbf{p}) = -2\mathbf{J}(\mathbf{p})^T \mathbf{\Gamma} [\hat{\mathbf{y}} - \mathbf{y}(\mathbf{p})] - \alpha \mathbf{g}_{log}(\mathbf{p}) + \beta \mathbf{g}_{reg}(\mathbf{p}) \quad (3.99)$$

where $\mathbf{J}(\mathbf{p}) = \left[\frac{\partial y_q(\mathbf{p})}{\partial p_r} \right]$ for $q = 1, \dots, N_{eq}$, $r = 1, \dots, N_{par}$ is the Jacobian matrix [55] of $\mathbf{y}(\mathbf{p})$,

$$\mathbf{g}_{log}(\mathbf{p}) = \begin{bmatrix} p_1^{-1} & \dots & p_{N_{par}}^{-1} \end{bmatrix}^T \quad (3.100)$$

and

$$\mathbf{g}_{reg}(\mathbf{p}) = \begin{bmatrix} 0 & \dots & 0 & 1 & \dots & 1 \end{bmatrix}^T. \quad (3.101)$$

The Hessian of $C_{bf}(\mathbf{p})$ is

$$\mathbf{H}(\mathbf{p}) = 2\mathbf{J}(\mathbf{p})^T \mathbf{\Gamma} \mathbf{J}(\mathbf{p}) - 2 \sum_{q=1}^{N_{eq}} w_q [\hat{y}_q - y_q(\mathbf{p})] \mathbf{G}_q(\mathbf{p}) + \alpha \mathbf{H}_{log}(\mathbf{p}) \quad (3.102)$$

where $\mathbf{G}_q(\mathbf{p}) = \left[\frac{\partial^2 y_q(\mathbf{p})}{\partial p_r \partial p_v} \right]$ for $r, v = 1, \dots, N_{par}$ is the Hessian of the function $y_q(\mathbf{p})$ and

$$\mathbf{H}_{log}(\mathbf{p}) = \text{diag} \left(\begin{bmatrix} p_1^{-2} & \dots & p_{N_{par}}^{-2} \end{bmatrix} \right). \quad (3.103)$$

We recall that the solution for $\Delta \mathbf{p}$ found by solving (3.95) is valid only if the Hessian $\mathbf{H}(\mathbf{p}_0)$ is positive definite on $\text{Null}(\mathbf{A}_{eq})$. Let us assume now that the update for the parameter vector is computed using (3.93) with $\Delta \mathbf{p}$ found by solving (3.95). Using the analysis form [56, Ch. 10], we have for a sufficiently small η

$$C_{bf}(\mathbf{p}_0 + \eta \Delta \mathbf{p}) \approx C_{bf}(\mathbf{p}_0) - \eta \Delta \mathbf{p}^T \mathbf{H}(\mathbf{p}_0) \Delta \mathbf{p} \quad (3.104)$$

where $\Delta \mathbf{p} \in \text{Null}(\mathbf{A}_{eq})$. We see that if $\mathbf{H}(\mathbf{p}_0)$ is positive definite on $\text{Null}(\mathbf{A}_{eq})$ then $\Delta \mathbf{p}$ found from (3.95) is a descent direction for $C_{bf}(\mathbf{p})$. In other words, for a sufficiently small η we can achieve $C_{bf}(\mathbf{p}) \leq C_{bf}(\mathbf{p}_0)$. On the other hand, if $\mathbf{H}(\mathbf{p}_0)$ is indefinite on $\text{Null}(\mathbf{A}_{eq})$ then $\Delta \mathbf{p}$ found from (3.95) is not necessarily a descent direction for $C_{bf}(\mathbf{p})$ and we may not be able to achieve $C_{bf}(\mathbf{p}) \leq C_{bf}(\mathbf{p}_0)$ for any η . We see that the second term on the right hand side of (3.102) is in general indefinite and hence, $\mathbf{H}(\mathbf{p})$ is also indefinite. Hence, using $\Delta \mathbf{p}$ computed from (3.95) where $\mathbf{H}(\mathbf{p})$ is computed using (3.102) does not lead to a convergent algorithm. One way for dealing with this problem is to use modified Newton's methods, where indefinite Hessian $\mathbf{H}(\mathbf{p})$ is replaced with a positive definite matrix, which is close to the original Hessian in certain sense. Some methods for doing this can be found in [55, sec. 4.4.2]. An obvious way to do this in our application is to replace the Hessian $\mathbf{H}(\mathbf{p})$ with its positive definite part

$$[\mathbf{H}(\mathbf{p})]_{pd} = 2\mathbf{J}(\mathbf{p})^T \mathbf{\Gamma} \mathbf{J}(\mathbf{p}) + \alpha \mathbf{H}_{log}(\mathbf{p}). \quad (3.105)$$

The step $\Delta \mathbf{p}$ is computed using (3.95) where $\mathbf{H}(\mathbf{p})$ is replaced with its $[\mathbf{H}(\mathbf{p})]_{pd}$. With positive definite Hessian on $\text{Null}(\mathbf{A}_{eq})$ the computed step $\Delta \mathbf{p}$ represents a descent direction and thus, leads to a convergent algorithm.

Finally, we need to select the step size η in (3.93). If the quadratic approximation in (3.92) holds exactly we should use $\eta = 1$ because then the optimization problem is solved in one step. In general, (3.92) holds only approximately and hence, using $\eta = 1$ to compute the updated parameter vector may not lead to $C_{bf}(\mathbf{p}) \leq C_{bf}(\mathbf{p}_0)$. Also the logarithmic barrier function is only approximated in (3.92) and hence, using $\eta = 1$ may violate the nonnegativity constraints. Therefore, we need a procedure for selection of the step size η in (3.93). This procedure is also known as line search [56]. Here, we use a very simple line search where we compute the updated parameter vector as

$$\mathbf{p} = \mathbf{p}_0 + \mu^v \Delta \mathbf{p} \quad (3.106)$$

where $0 < \mu < 1$ and $v = 0, 1, 2, \dots, V_{at}$. We start with $v = 0$ and keep increasing v until all nonnegativity constraints are satisfied and $C_{bf}(\mathbf{p}) < C_{bf}(\mathbf{p}_0)$. These conditions can be satisfied for a sufficiently small step length since $\Delta \mathbf{p}$ is a descent direction for $C_{bf}(\mathbf{p})$. The line search procedure is the final part of the algorithm.

Let us summarize the algorithm for solving (3.86). Starting from a random initial point, which satisfies all constraints from (3.86), we iteratively update the parameter vector \mathbf{p} defined by (3.89). In each iteration we compute $\mathbf{J}(\mathbf{p})$ as partial derivatives of model equations (3.36), $\mathbf{H}(\mathbf{p})$ using (3.105), and $\mathbf{g}(\mathbf{p})$ using (3.99). Then we compute $\Delta\mathbf{p}$ by solving equations (3.95) and update the parameter vector using line search (3.106). The iterations are repeated until convergence. The algorithm converges to a local minimum of the criterion function $C_{bf}(\mathbf{p})$. The global minimum can be found from the results obtained by running the algorithm from different initial points.

Now we recall that the problem (3.86) is a simplification of our original problem defined in (3.86). Using the computed solution of (3.86) we would like to find the solution of the original problem. Let $\hat{\mathbf{p}}_0$ be the obtained solution of (3.86) and let $\hat{\mathbf{F}}_0$, $\hat{\mathbf{C}}_0$, and $\hat{\mathbf{K}}_0$ be the corresponding matrices found from $\hat{\mathbf{p}}_0$ using (3.89) and (3.87). The solution $\hat{\mathbf{p}}_0$ is computed for the regularization parameter β chosen so that $C_{wls}(\hat{\mathbf{p}}_0) \gg C_{reg}(\hat{\mathbf{p}}_0)$. It is important to understand that for $\beta = 0$ the problem (3.86) may have a continuum of solutions instead of a unique solution. Since our numerical algorithm is a modification of Newton's method it can work only if the Hessian is nonsingular at the solution. If there is a continuum of solutions the Hessian is singular around any such solution point and hence, applying our algorithm would be problematic. In order to avoid this problem we had to use $\beta > 0$ which assures that the solution is unique. In order to identify zero entries of $\hat{\mathbf{K}}_0$ we use the following iterative procedure. We find the entry in $\hat{\mathbf{K}}_0$ with the smallest absolute value and compute the new solution of (3.86) where this entry in \mathbf{K} is constrained to be zero, $\hat{\mathbf{p}}_0$ is the initial point and the parameter β chosen so that the influence of the regularization term is negligible. This solution is found using the iterative algorithm described above. Let this new solution be $\hat{\mathbf{p}}_1$. If

$$\frac{C_{wls}(\hat{\mathbf{p}}_1) - C_{wls}(\hat{\mathbf{p}}_0)}{C_{wls}(\hat{\mathbf{p}}_0)} < \gamma \quad (3.107)$$

where γ is a suitable threshold, we set $\hat{\mathbf{p}}_0 = \hat{\mathbf{p}}_1$ and compute the corresponding matrices $\hat{\mathbf{F}}_0$, $\hat{\mathbf{C}}_0$, and $\hat{\mathbf{K}}_0$. In the second iteration we find the entry with the smallest absolute value on the subset of the entries $\hat{\mathbf{K}}_0$ which have not been tested yet and repeat the same procedure. This testing procedure is repeated until all entries of $\hat{\mathbf{K}}_0$ have been

tested.

The algorithm for solving (3.86) does not take into account that $c_{im} \in \{0, 1\}$. This constraint can be taken into account by an iterative algorithm where in each iteration, starting from some initial point, the existing estimates $\hat{\mathbf{F}}$ and $\hat{\mathbf{K}}$ would be updated using the algorithm for solving (3.86) and $\hat{\mathbf{C}}$ would be updated using the technique from [28]. However, this type of algorithm would require an accurate initial point for $\hat{\mathbf{C}}$ which is not available in our application. This is why we computed the estimates with imposing only $c_{im} \geq 0$. Let $\hat{\mathbf{F}}_0$, $\hat{\mathbf{C}}_0$, and $\hat{\mathbf{K}}_0$ be the estimates obtained after solving (3.86) and the testing procedure for determining zero entries in \mathbf{K} . In the next step we project the entries of $\hat{\mathbf{C}}_0$ onto $\{0, 1\}$ alphabet which is a simple clustering problem. Using this estimate $\hat{\mathbf{C}}$ with binary entries we compute the final estimates of \mathbf{F} and the nonzero entries of \mathbf{K}

$$[\hat{\mathbf{F}}, \hat{\mathbf{K}}] = \underset{\mathbf{F} \geq 0, \mathbf{K}, \mathbf{1}^T \mathbf{F} = \mathbf{1}^T}{\operatorname{argmin}} C_{wls}(\mathbf{F}, \hat{\mathbf{C}}, \mathbf{K}) \quad (3.108)$$

which can be solved using the algorithm developed for solving (3.86) with the initial point $\hat{\mathbf{F}}_0$ and $\hat{\mathbf{K}}_0$.

Let us summarize the algorithm for computing $\hat{\mathbf{F}}$, $\hat{\mathbf{C}}$, and $\hat{\mathbf{K}}$ from $\hat{\mathbf{Y}}$. First, we obtain our initial estimates by solving (3.86). Second, we determine zero entries in \mathbf{K} using the described testing procedure. Third, we project the entries of the estimate of \mathbf{C} onto $\{0, 1\}$ alphabet and compute the final estimates for \mathbf{F} and \mathbf{K} by solving (3.108).

Let us assume that we have computed $\hat{\mathbf{C}}$ from $\hat{\mathbf{Y}}$. Now it is possible to find power spectra of the received signals and noise. Let $\hat{\mathbf{Z}}$ be a J_{fb} by I_{cl} matrix where the i -th column of $\hat{\mathbf{Z}}$ is a power spectrum estimate computed from the received samples belonging to the i -th cluster. We define the criterion function

$$C_{wls}^{(so)} = \sum_{j=1}^{J_{fb}} \sum_{i=1}^{I_{cl}} \gamma_i \left(\hat{\mathbf{Z}}(j, i) - \sum_{m=1}^M g_{jm} c_{im} - n_j \right)^2 \quad (3.109)$$

where the weights γ_i are the same as those in (3.82). The parameters of interest can be estimated by solving

$$[\hat{\mathbf{G}}, \hat{\mathbf{n}}] = \underset{\mathbf{G} \geq 0, \mathbf{n} \geq 0}{\operatorname{argmin}} C_{wls}^{(so)}(\mathbf{G}, \mathbf{n}, \hat{\mathbf{C}}) \quad (3.110)$$

where $\hat{\mathbf{C}}$ has been estimated from $\hat{\mathbf{Y}}$. The problem (3.110) can be solved using the same method that was used for solving (3.86).

3.4 Numerical examples

We illustrate the proposed algorithm with three simulation examples. In all three examples we consider a setup with one sensor and two sources, whose locations are shown in Figure 3.9. Again, the channels between the sensor and the sources are transfer functions measured in the ORBIT room in WINLAB for the setup shown in Figure 3.9 [66]. Thus, using this technique the real world scenarios are mapped in the simulations.

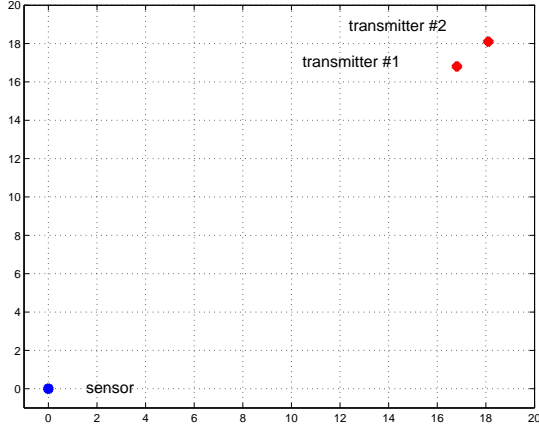


Figure 3.9: Locations of the sensor(blue) and sources(red). Distances are in meters.

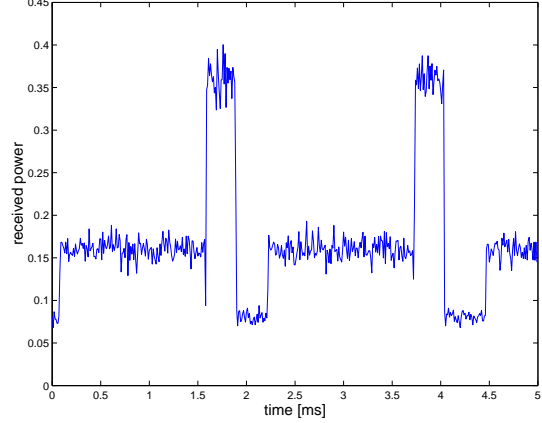


Figure 3.10: Power trace of the received signal computed with the time resolution $T = 10\mu s$

Example 1. Both sources are transmitting DBPSK signals with Barker sequence spreading used in 802.11b systems [67]. The sources transmit with equal power in the same channel, which is observed by the sensor. Total observed bandwidth is $W = 20MHz$, the sampling period is $T_s = 0.05\mu s$. Figure 3.10 shows power trace of the received signal at the sensor, where each point is the average power computed over $N_T = 200$ consecutive samples, which corresponds to the interval of $T = N_T T_s = 10\mu s$. Figure 3.11 shows spectrogram of the received signal computed with the same time resolution of $T = 10\mu s$. The total observation time on this example is 5ms. We see

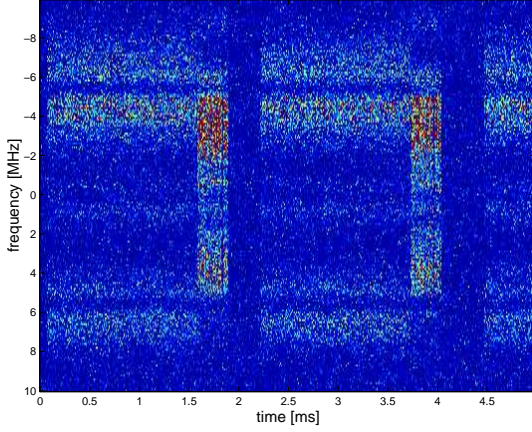


Figure 3.11: Spectrogram of the received signal computed with the time resolution $T = 10\mu s$

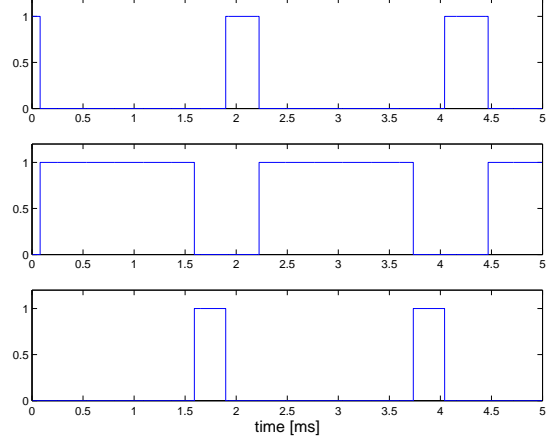


Figure 3.12: Segmentation sequences estimated by applying the segmentation algorithm on the received signal

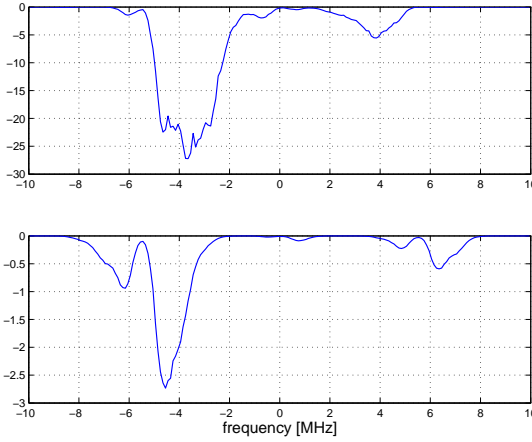


Figure 3.13: Diagonal slices of the recovered fourth order spectra

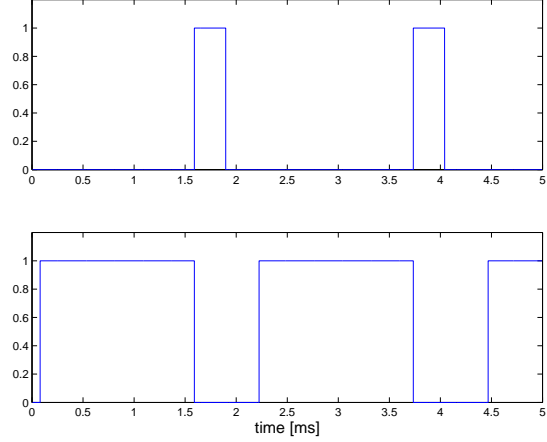


Figure 3.14: Recovered activity sequences

a typical 802.11 traffic: one source is sending packets and the other one is replying with acknowledgments. Received signal at the sensor is corrupted with additive white Gaussian noise. We define SNR for each source-sensor pair as the ratio of the average received source signal power (when the source is on) at the sensor and the average sensor noise power. The SNR values in the example shown in Figure 3.10 are 0 and 5.4 dB. First, we apply our segmentation algorithm on the received signal which recovers the segmentation sequences shown in Figure 3.12. The received signal consists of segments

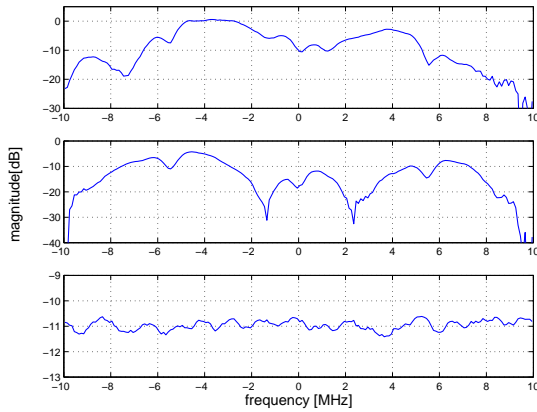


Figure 3.15: Recovered power spectra

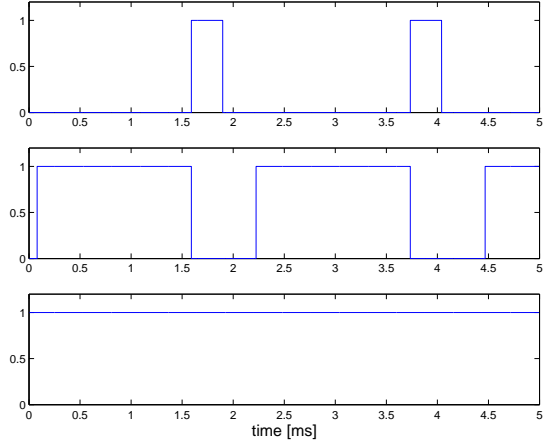
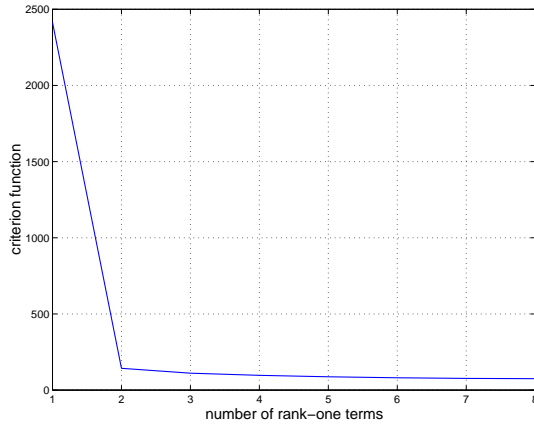
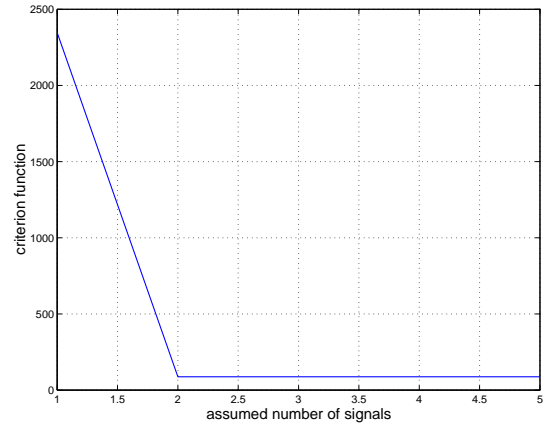


Figure 3.16: Recovered activity sequences

Figure 3.17: Criterion function versus the parameter R for $M = 2$ Figure 3.18: Criterion function versus the parameter M for $R = 5$

where each segment contains one of three different types of signals. The top graph shows the sequence containing segments with sensor noise only, the middle graph shows the sequence containing segments with the first received signal plus sensor noise and the bottom graph shows the sequence containing segments with the second received signal plus sensor noise. Using these segmentation sequences we estimate the three-way array $\hat{\mathbf{Y}}$ where $J_{fb} = 200$ and $I_{cl} = 3$. Now we wish to apply the proposed computational algorithm on $\hat{\mathbf{Y}}$. But before we can do that we need to determine the parameters M and R . In order to understand the effect of R we fix $M = 2$ and apply our algorithm

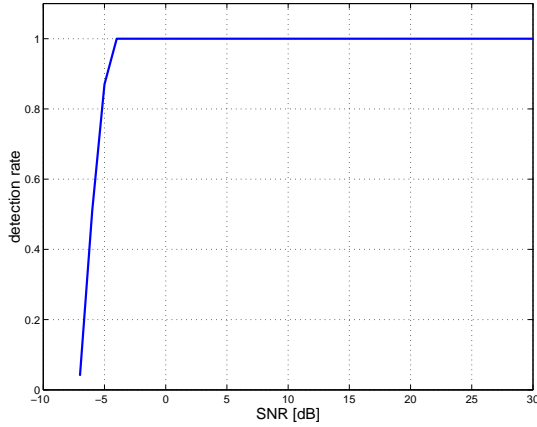


Figure 3.19: Detection rate of the correct number of clusters

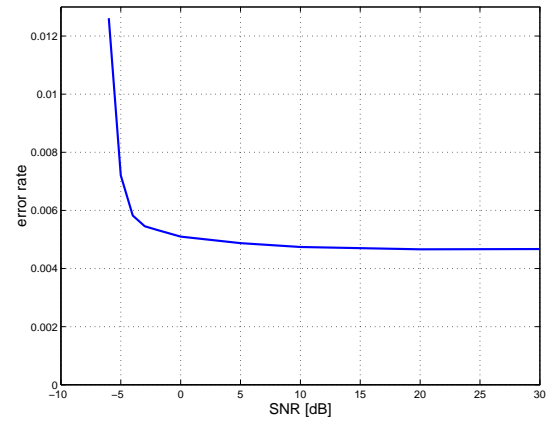


Figure 3.20: Segmentation error rate

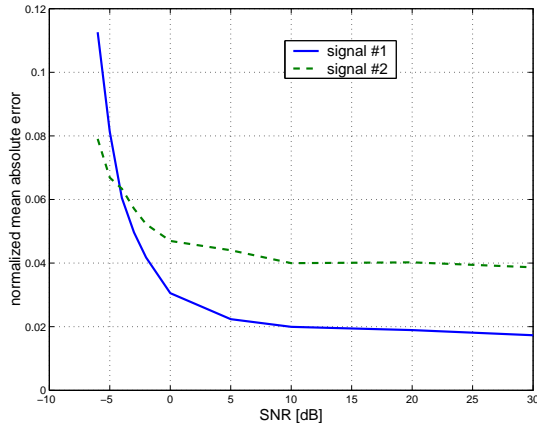


Figure 3.21: Normalized mean absolute errors for estimates of the columns of \mathbf{F}

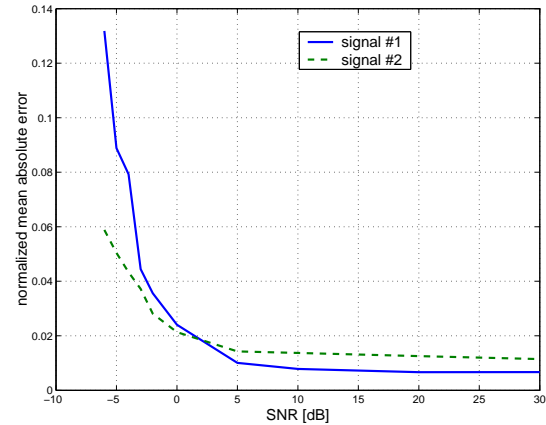


Figure 3.22: Normalized mean absolute errors for estimates of the columns of \mathbf{K}

for different values of R . Figure 3.17 shows the minimum value of the criterion function C_{wls} versus R . We see that the curve in Figure 3.17 becomes very flat when $R > 2$. Next, we fix $R = 5$ and apply our algorithm for different values of M . Figure 3.18 shows the minimum value of the criterion function C_{wls} versus M . We see that the curve in Figure 3.18 becomes almost completely flat when $M > 2$, which implies that $M = 2$ is a reasonable estimate. From the curve in Figure 3.17 we see that $R = 2$ is a reasonable estimate. We know that $M = 2$ and $R = 2$ are correct estimates because in this example there are two linearly modulated signals. We see from the curve in Figure 3.18 that the parameter M can be correctly estimated without knowing the true value of R . Similarly,

the parameter R can be estimated from the curve showing the minimum values of the criterion function versus R for $M = R$ which is very similar to the curve shown in Figure 3.17. Thus, the parameter R can be estimated without knowing the true value of M . In this example the estimates for M and R are obtained by visual inspection of the curves showing the minimum values of the criterion function versus M and R . In order to obtain a fully automated algorithm it is necessary to develop numerical criteria for estimation of M and R from the above mentioned curves. These numerical criteria must determine the points when the curves shown in Figure 3.18 and 3.17 become sufficiently flat. Thus, such criteria would determine if any given signal or any of its rank-one terms can be detected or not. Obviously, if a signal is sufficiently weak its effect on the criterion function is negligible and hence, it cannot be detected. We leave the problem of designing these numerical criteria for future work. As mentioned earlier the parameters M and R can be estimated as effective ranks of the matrices from (3.53) and (3.58) where \mathbf{Y} is replaced with $\hat{\mathbf{Y}}$. However, this approach does not take into account any application specific constraints on the terms into which $\hat{\mathbf{Y}}$ is decomposed. On the other hand, the curves showing the minimum values of the criterion function versus M and R take into account any application specific constraints and hence, this approach is more appropriate here. A similar approach was used for some other model order selection problems [95]. Another parameter needed for the computational algorithm is the threshold γ from (3.107). The value of this parameter is chosen from the curve shown Figure 3.17. Let us denote this curve with $C_{min}(R)$ and let \hat{R} be the estimated value of R . Then we select $\gamma = (C_{min}(\hat{R}) - C_{min}(\hat{R} + 1))/C_{min}(\hat{R})$.

Applying the proposed computational algorithm on $\hat{\mathbf{Y}}$ with $M = 2$ and $R = 2$ we estimate fourth order spectrum slices for each of the signals and the corresponding activity sequences. Figure 3.13 shows diagonal entries of the estimated fourth order spectrum slices. These diagonal entries are kurtosis values at different frequency bins. Figure 3.14 shows the estimated activity sequences. We see that the contributions of the two signals have been correctly recovered. Since the sensor noise is Gaussian it does not contribute to $\hat{\mathbf{Y}}$ any terms that can be modeled. However, the presence of noise affects the estimated parameters because it increases the estimation errors in $\hat{\mathbf{Y}}$. Next, we

estimate the power spectra matrix $\hat{\mathbf{Z}}$ where $J_{fb} = 200$ and $I_{cl} = 3$. Using the recovered activity sequences from $\hat{\mathbf{Y}}$ we estimate power spectra for each of the received signals and noise. Figure 3.15 shows the estimated power spectra and Figure 3.16 shows the corresponding activity sequences. We see from Figure 3.15 that the estimated power spectra of the two received signals are very different even though the two signals use the same modulation format. These differences are caused by the frequency selective channels between the transmitters and the sensor. Sensor noise contributes its own rank one term to $\hat{\mathbf{Z}}$. The estimated noise power spectrum is relatively flat, which was expected because the sensor noise was white in this example.

Next, we study behavior of the algorithm for a range of SNR values. At each SNR point we perform 100 runs. In each run we generate new realizations of the signals and noise, apply our segmentation algorithm on the received signal, estimate $\hat{\mathbf{Y}}$ and compute $\hat{\mathbf{F}}$, $\hat{\mathbf{C}}$, and $\hat{\mathbf{K}}$ from $\hat{\mathbf{Y}}$. In order to asses performance of the segmentation algorithm at each SNR point we compute the percentage of runs when the true number of clusters is detected. The results are shown in Figure 3.19. We see that the algorithm works up to the SNR value when the two nearest clusters merge and cannot be resolved any more. Figure 3.20 shows the segmentation error rate. We see that the rate is approximately constant for sufficiently high SNR values. When SNR approaches the threshold value the two nearest clusters start to have a significant overlap which causes a steep increase in the observed error rate. In order to asses performance of the algorithm for computing $\hat{\mathbf{F}}$, $\hat{\mathbf{C}}$, and $\hat{\mathbf{K}}$ from $\hat{\mathbf{Y}}$ we use normalized mean absolute estimation error. Let us explain how this error is computed for the estimate of the first column of $\mathbf{F} = \begin{bmatrix} \mathbf{f}_1 & \mathbf{f}_2 \end{bmatrix}$. Let $\hat{\mathbf{f}}_1^{(n)}$ for $n = 1, \dots, N$ be the estimates of \mathbf{f}_1 obtained at one SNR value for N different realizations of the signals and noise. The normalized mean absolute estimation error for this subset of parameters is defined as $\sum_{n=1}^N \|\hat{\mathbf{f}}_1^{(n)} - \mathbf{f}_1\|_1 / (N\|\mathbf{f}_1\|_1)$ where \mathbf{f}_1 contains true values of the estimated parameters. Figure 3.21 shows the estimation errors for the estimates of the columns of \mathbf{F} . At high SNR values the estimation errors cannot fall below some floor values because these estimates are obtained from $\hat{\mathbf{Y}}$ which always contains some amount of error. These errors in $\hat{\mathbf{Y}}$ are caused by estimation from a finite sample size and they exist even if the SNR is very high. When the SNR

approaches the threshold value the estimates of the entries of $\hat{\mathbf{Y}}$ become less and less accurate which causes a steep increase of the estimation errors shown in Figure 3.21. Figure 3.22 shows the estimation errors for the estimates of the columns of \mathbf{K} . We see that these estimates behave similarly to the estimates of the columns of \mathbf{F} .

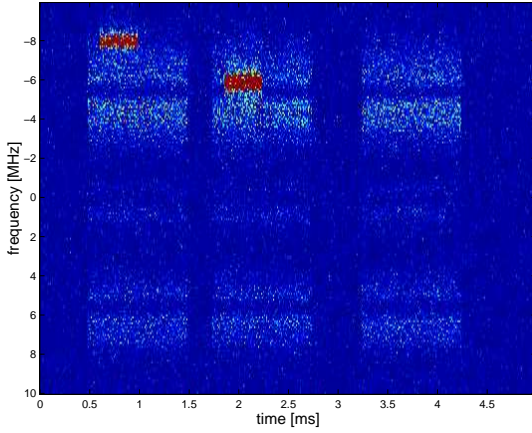


Figure 3.23: Spectrogram of the received signal computed with the time resolution $T = 10\mu s$

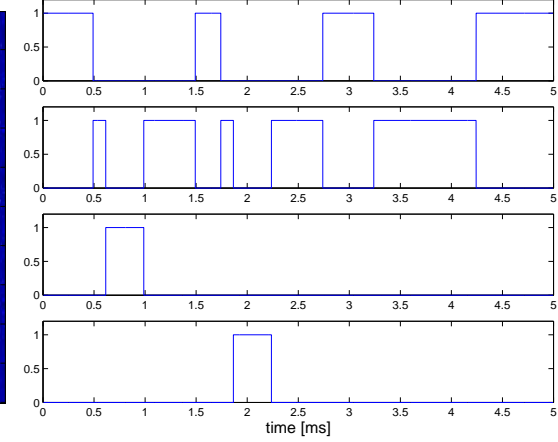


Figure 3.24: Segmentation sequences estimated by applying the segmentation algorithm on the received signal

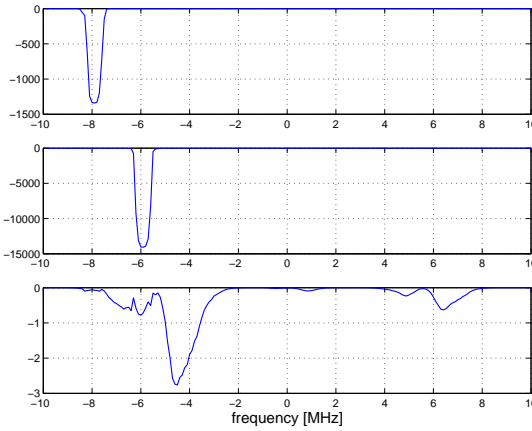


Figure 3.25: Diagonal slices of the recovered fourth order spectra

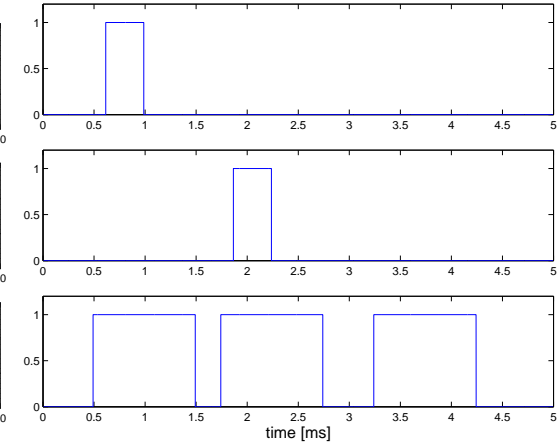


Figure 3.26: Recovered activity sequences

Example 2. In this example the transmitter nearer to the sensor is transmitting DBPSK signals with Barker sequence spreading and the other transmitter is transmitting GFSK signal with frequency hopping used in Bluetooth systems [96]. Each

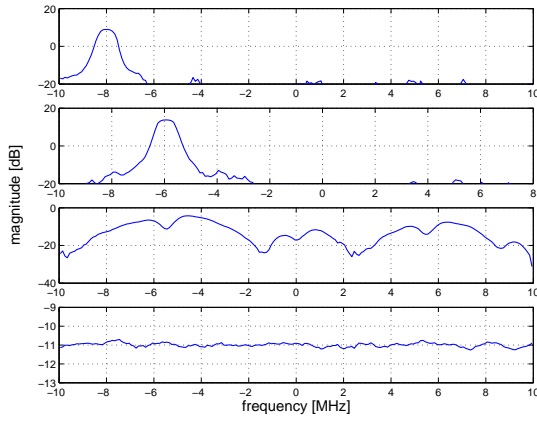


Figure 3.27: Recovered power spectra

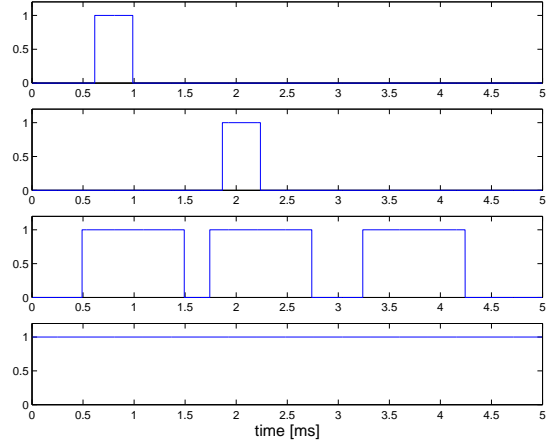
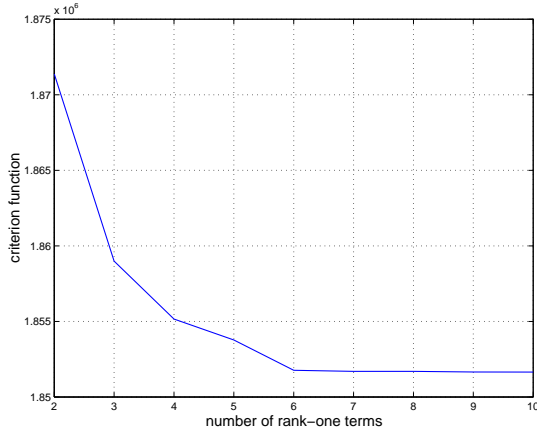
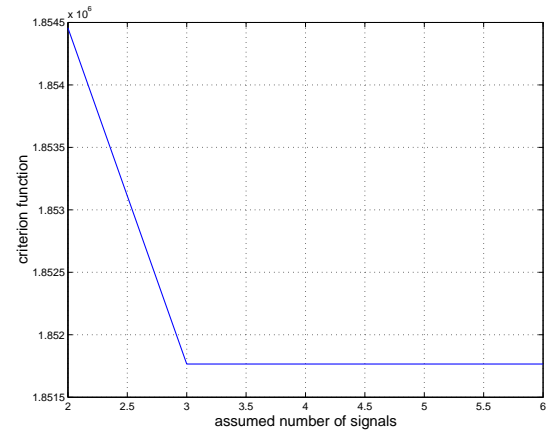


Figure 3.28: Recovered activity sequences

Figure 3.29: Criterion function versus the parameter R for $M = 3$ Figure 3.30: Criterion function versus the parameter M for $R = 6$

Bluetooth packet is transmitted over one of 79 different channels, where each channel is approximately 1 MHz wide. The spectrogram of the received signal computed with time resolution of $T=10\mu s$ is shown in Figure 3.23. The sensor observes one 20 MHz wide channel over the total observation time of 5ms. During this time the DBPSK transmitter transmits three and the Bluetooth transmitter transmits two packets. Both Bluetooth packets collide with 802.11b packets as can be seen in Figure 3.23. The SNR for the DBPSK signal is 0 dB. The two Bluetooth packets are transmitted over two different 1 MHz wide channels. Thus, the Bluetooth transmitter uses two different signals

in the observed time-frequency slot. Each of these two signals uses GFSK, which is a form of frequency modulated signal with Gaussian pulse shaping [97]. The SNR values for the two GFSK signals are 5.7 and 10.5 dB. This difference in SNR values is caused by the frequency selective channel between the Bluetooth transmitter and the sensor. First, we apply our segmentation algorithm on the received signal which recovers the four segmentation sequences shown in Figure 3.24. Hence, $I_{cl} = 4$ for this example. The top sequence in Figure 3.24 contains the segments with the sensor noise only, the second sequence contains the segments with the DBPSK signal and sensor noise, the third sequence contains the segment with the first GFSK signal, the DBPSK signal, and the sensor noise and the fourth sequence contains the segment with the second GFSK signal, the DBPSK signal, and the sensor noise. Using the estimated segmentation sequences we estimate $\hat{\mathbf{Y}}$ with $J_{fb} = 200$ and $I_{cl} = 4$. Next, we need to determine the parameters M and R . First, we fix $M = 3$ and apply the computational algorithm on $\hat{\mathbf{Y}}$ for different values of R . Figure 3.29 shows the minimum value of the criterion function C_{wls} for different values of R . We see that for $R > 6$ the curve in Figure 3.29 becomes very flat. In the second experiment, we fix $R = 6$ and apply the computational algorithm on $\hat{\mathbf{Y}}$ for different values of M . Figure 3.30 shows the minimum value of the criterion function C_{wls} for different values of M . We see that for $M > 3$ the curve in Figure 3.30 becomes very flat. From these two experiments we see that $M = 3$ and $R = 6$ are reasonable estimates of these two parameters. The estimate $M = 3$ is correct because there is one DBPSK signal and two GFSK signals in the observed time-frequency slot. Applying the computational algorithm on $\hat{\mathbf{Y}}$ with $M = 3$ and $R = 6$ recovers the fourth order spectrum slices whose diagonal entries are shown in Figure 3.25 and the activity sequences shown in Figure 3.26. The top graphs in these two figures correspond to the first GFSK signal, the middle graphs correspond to the second GFSK signal and the bottom graphs correspond to the DBPSK signal. The estimated matrix $\hat{\mathbf{K}}$ is of the block form (3.32) with $R_1 = 2$, $R_2 = 3$, and $R_3 = 1$. Since $R_3 = 1$ the recovered fourth order spectrum of the DBPSK signal is a rank-one matrix. This makes sense because the DBPSK signal is linearly modulated which implies that its fourth order spectrum slice is also in theory a rank-one matrix. Since $R_1 = 2$ and $R_2 = 3$ the recovered fourth

order spectrum of the first GFSK signal is a rank-two matrix and the recovered fourth order spectrum of the second GFSK signal is a rank-three matrix. In general, GFSK signal is nonlinear and its fourth order spectrum slice is not a rank-one matrix. In this example, the spectrum of the first GFSK signal is approximated with a rank-two matrix and the spectrum of the second GFSK signal is approximated with a rank-three matrix. In theory, the ranks of two matrices are higher than two or three but these additional rank-one terms are negligible in this example. This can be seen from Figure 3.29 where increasing R beyond $R_1 + R_2 + R_3 = 6$ does not cause any significant drop in the minimum value of the criterion function. The SNR of the second GFSK signal is significantly higher than that of the first GFSK signal. This explains why the recovered spectrum of the first signal is a rank-two matrix and the recovered spectrum of the second signal is a rank-three matrix. Finally, we estimate the power spectra matrix $\hat{\mathbf{Z}}$ with $J_{fb} = 200$ and $I_{cl} = 4$. Using the recovered activity sequences from $\hat{\mathbf{Y}}$ we estimate power spectra for each of the received signals and noise. Figure 3.27 shows the estimated power spectra and Figure 3.28 shows the corresponding activity sequences. We see that the contributions of different signals have been recovered correctly despite the collisions between the signals. In this example the SNR values were fixed. When observed over a range of SNR values the algorithm exhibits the same type of threshold behavior as in the first example. If the SNR values are sufficiently small so that the two nearest clusters cannot be resolved any more the algorithm breaks down. In this example the two nearest clusters are the one containing time intervals with noise only and the one containing time intervals with the DBPSK signal plus noise. Since this situation is the same as in the first example the threshold SNR value of the DBPSK signal at which the algorithm breaks down is the same as in the first example (around $-5dB$).

Example 3. In this example the transmitter nearer to the sensor is transmitting DBPSK signals with Barker sequence spreading and the other is transmitting OFDM signal used in 802.11a/g systems [67]. The sensor observes one 20 MHz wide channel over the observation interval of $5ms$. The SNR values are 0 for the DBPSK signal and 6.7 dB for the OFDM signal. The transmitted packets from the two sources are

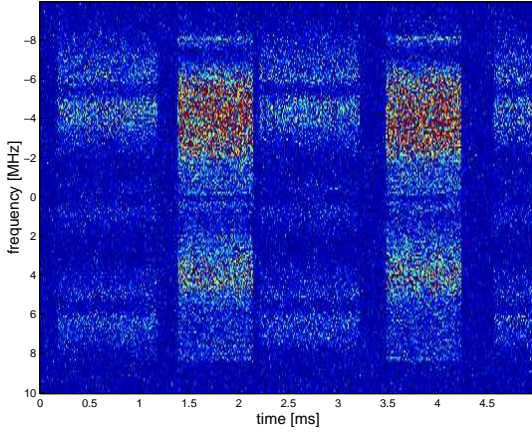


Figure 3.31: Spectrogram of the received signal computed with the time resolution $T = 10\mu s$

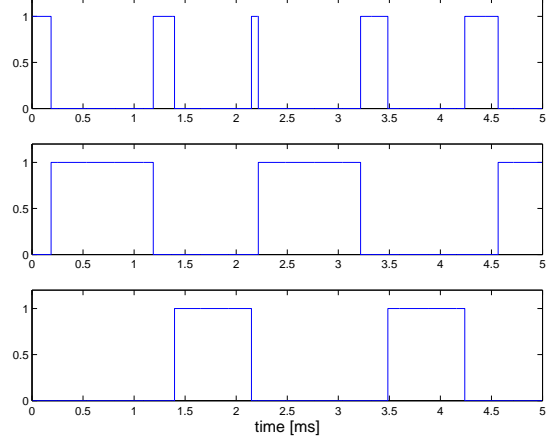


Figure 3.32: Segmentation sequences estimated by applying the segmentation algorithm on the received signal

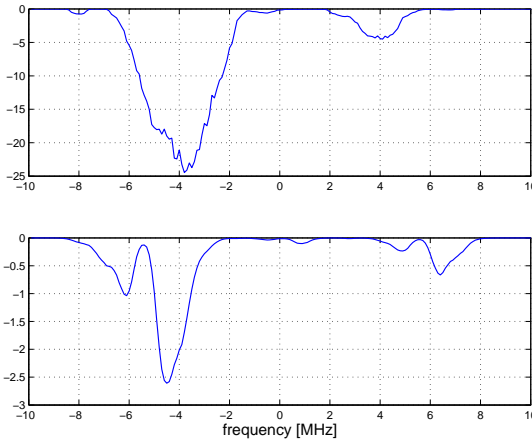


Figure 3.33: Diagonal slices of the recovered fourth order spectra

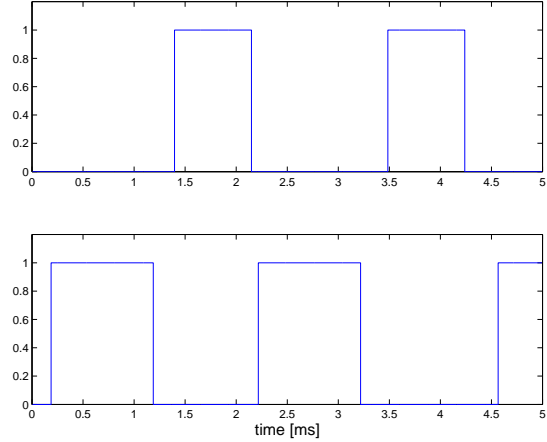


Figure 3.34: Recovered activity sequences

interleaved in time and there are no collisions as can be seen from the spectrogram shown in Figure 3.31. First, we apply our segmentation algorithm on the received signal which recovers the three segmentation sequences shown in Figure 3.32. Hence, $I_{cl} = 3$ for this example. The top sequence in Figure 3.32 contains the segments with the sensor noise only, the second sequence contains the segments with the DBPSK signal and sensor noise, and the third sequence contains the segment with the OFDM signal and the sensor noise. Using the estimated segmentation sequences we estimate $\hat{\mathbf{Y}}$ with

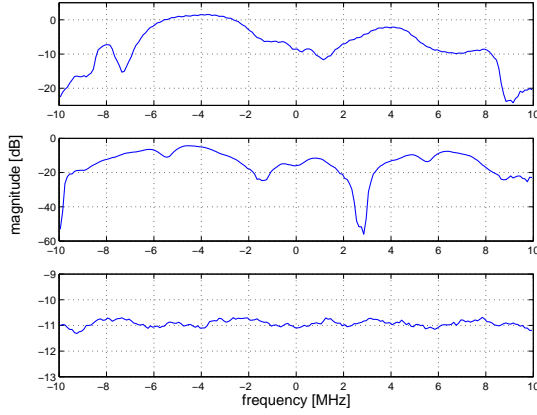


Figure 3.35: Recovered power spectra

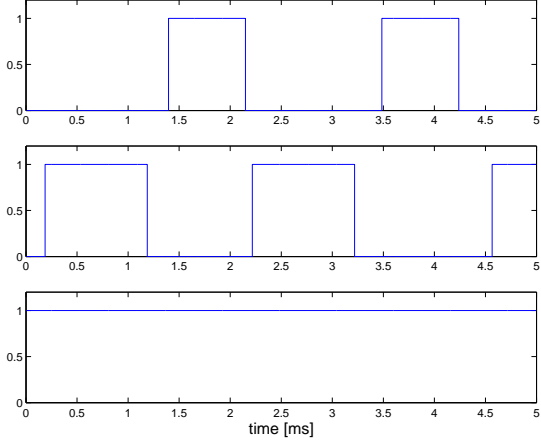
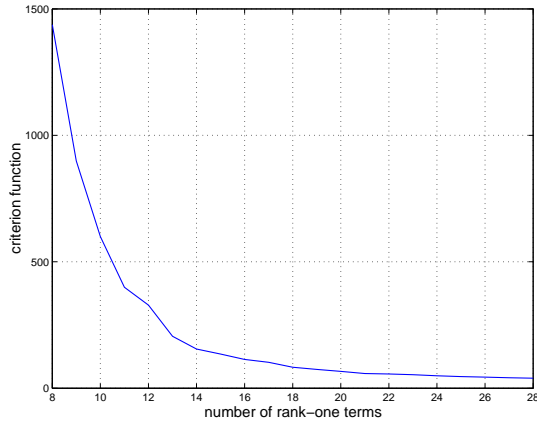
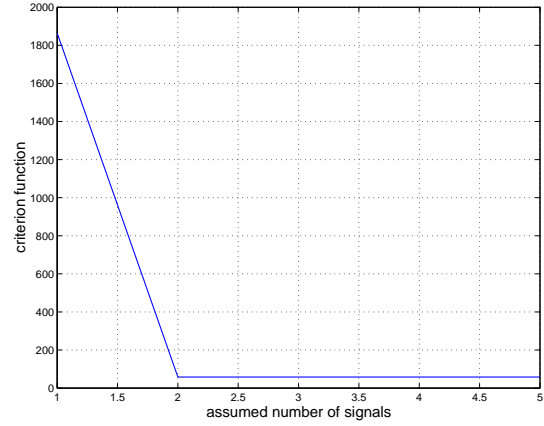


Figure 3.36: Recovered activity sequences

Figure 3.37: Criterion function versus the parameter R for $M = 2$ Figure 3.38: Criterion function versus the parameter M for $R = 21$

$J_{fb} = 200$ and $I_{cl} = 3$. Next, we need to determine the parameters M and R . First, we fix $M = 2$ and apply the computational algorithm on $\hat{\mathbf{Y}}$ for different values of R . Figure 3.37 shows the minimum value of the criterion function C_{wls} for different values of R . In the second experiment, we fix $R = 21$ and apply the computational algorithm on $\hat{\mathbf{Y}}$ for different values of M . Figure 3.38 shows the minimum value of the criterion function C_{wls} for different values of M . We see that for $M > 2$ the curve in Figure 3.38 becomes very flat. The estimate $M = 2$ is correct because there are two signals in this example. Applying the computational algorithm on $\hat{\mathbf{Y}}$ with $M = 2$ and $R = 21$

recovers the fourth order spectrum slices whose diagonal entries are shown in Figure 3.33 and the activity sequences shown in Figure 3.34. The top graphs in these two figures correspond to the OFDM signal and the bottom graphs correspond to the DBPSK signal. The estimated matrix $\hat{\mathbf{K}}$ is of the block form (3.32) with $R_1 = 20$ and $R_2 = 1$. Since $R_2 = 1$ the recovered fourth order spectrum of the DBPSK signal is a rank-one matrix. Since $R_1 = 20$ the recovered fourth order spectrum of the OFDM signal is a matrix of the rank twenty. This type of OFDM signal consists of 52 carriers where each carrier is one linearly modulated signal. Hence, the fourth order spectrum slice of the OFDM signal should be of the rank 52. However, the curve shown in Figure 3.37 drops very slowly for $R > 21$. This can be explained as follows. Different OFDM carriers are attenuated differently by the frequency selective channel. On the other hand, $\hat{\mathbf{Y}}$ is estimated from a finite sample size and it always contains a some amount of error. The contributions to $\hat{\mathbf{Y}}$ of the OFDM carriers which are not attenuated significantly by the channel is much larger then the estimation error contained in $\hat{\mathbf{Y}}$ and they can be easily detected. However, some OFDM carriers are attenuated significantly by the channel and their contributions to $\hat{\mathbf{Y}}$ are of the same order of magnitude as the estimation error contained in $\hat{\mathbf{Y}}$ and hence, they are difficult to detect. These facts explain why the rate of decrease of the curve in Figure 3.37 becomes smaller as R increases. In this example it is not trivial to estimate R because the rate of decrease of the curve in Figure 3.37 reduces gradually. One reasonable estimate is $R = 21$. We see that increasing R beyond 21 leads to very small decreases of the minimum value of the criterion function, which implies that any additional rank-one terms contributed by the OFDM signal are of the same order of magnitude as the estimation error contained in $\hat{\mathbf{Y}}$ and thus, they cannot be detected. Finally, we estimate the power spectra matrix $\hat{\mathbf{Z}}$ with $J_{fb} = 200$ and $I_{cl} = 3$. Using the recovered activity sequences from $\hat{\mathbf{Y}}$ we estimate power spectra for each of the received signals and noise. Figure 3.35 shows the estimated power spectra and Figure 3.36 shows the corresponding activity sequences. We note that in this example the recovered fourth order spectrum slices reveal the difference between the signal with single carrier modulation and the signal with multicarrier modulation. This difference cannot be seen from the power spectra of these two signals. In this example

the SNR values were fixed. When observed over a range of SNR values the algorithm exhibits the same type of threshold behavior as in the previous examples. Again, the two nearest clusters are the one containing time intervals with noise only and the one containing time intervals with the DBPSK signal plus noise. Hence, the threshold SNR value of the DBPSK signal at which the algorithm breaks down is the same as in the previous two examples (around $-5dB$).

From these specific examples we can draw the following conclusion about performance of the proposed signal analysis algorithm in general. The algorithm consists of the segmentation step and the fourth order spectrum based analysis step. The algorithm has two main limitations. The segmentation algorithm works correctly while all clusters present in the received signal can be resolved. Usually, as SNR values decrease at some point the two nearest clusters become impossible to resolve and the algorithm breaks down. This is the first limitation of the algorithm. In the fourth order spectrum based analysis part we can detect only sufficiently strong signals. As a signal becomes weaker its influence on the criterion function reduces. As the influence on the criterion function reduces at some point the signal or some of its rank-one components become impossible to detect. This is the second limitation of the algorithm. These limitations can be overcome by using multiple sensors if available.

3.5 Conclusion and future work

In the previous chapter we proposed an algorithm for localizing in time statistically homogeneous segments in the received signal. In this chapter we have shown how spectra and the associated activity sequences can be estimated using fourth order spectra of the segments localized in the first step. Thus, using the proposed analysis method each individual signal can be localized in time and frequency. The proposed method requires only one sensor. As already noted some topics are left for future work. The method described in this chapter uses only partial information from fourth order spectrum. It is possible to extend the method so that it uses full information from fourth order spectrum. The second topic for future work is development of numerical algorithm for estimation of the number of signals and the number of the rank-one terms comprising

the signals.

The method proposed in this chapter is based on spectral analysis where the underlying signals are projected onto complex exponentials. This approach is very general and it can be applied to many types of real world signals. However, in some case this approach results in a large number of rank-one terms needed to represent the underlying signals and then the method may not be very practical. In those cases it is necessary to project the underlying signals onto different basis functions which produce simpler signal representations. Therefore, finding appropriate representations for real world communication signals is another interesting topic for future work.

Chapter 4

Algorithm with Multiple Sensors

In this chapter we assume that multiple sensors are available. First, we consider the scenario with multiple packet based transmitters and multiple sensors. Using the proposed segmentation algorithm and the fourth order spectrum based analysis it is possible to estimate for each transmitter its activity sequence in time and the received second and fourth order spectra at different sensors. We outline a distributed algorithm which uses observations from multiple sensors for estimation of these quantities. From the received second order spectra of each identified transmitted signal it is possible to estimate the magnitude frequency responses of the channels between the transmitter and the sensors and the transmitted signal spectrum. For realistic channels the formulated blind deconvolution problem is often ill conditioned and then the solutions obtained using existing algorithms are useless. We show that a useful solution can be obtained using the l_1 norm regularization and propose a numerical algorithm for the single input multiple output blind deconvolution. The estimated channel transfer function between the transmitter and the sensors depend on the transmitter location relative to the sensors. Therefore, multiple sensors enable transmitter localization, which is not possible using a single sensor.

4.1 Outline of the algorithm

Let us assume that there is a network of Q sensors observing a frequency band used by M packet based radio transmitters. Our goal is to develop a distributed algorithm for the sensor network for localization of packet based radio transmitters in space, time, and frequency. We proceed under two assumptions. First, we assume that there is

a dedicated control channel for communication between the sensors. Our second assumption that the sensors perform the measurements synchronously in time. Achieving time synchronization in wireless networks is a separate research problem which will not be considered here. Some of the available techniques for time synchronization achieve accuracy of about $1\mu s$ [98] [99]. Our algorithm consists of the following steps.

(1) All sensors observe the same frequency band over the same time window. Each sensor collects samples of the received signal over the observed time window. Received signal samples at the q -th sensor are $r_q(k)$ where $k = 1, \dots, K$ and $q = 1, \dots, Q$. At the q -th sensor we apply our segmentation algorithm on $r_q(k)$ which produces the segmentation matrix $\hat{\mathbf{S}}_q$ and the matrix \mathbf{D}_q which contains the pairwise distances between the identified clusters. The q -th sensor transmits $\hat{\mathbf{S}}_q$ and \mathbf{D}_q to all other sensors, which can be done using one broadcast message. This process is repeated for $q = 1, \dots, Q$. At each sensor we perform the fusion of $\hat{\mathbf{S}}_q$ for $q = 1, \dots, Q$ which produces the final segmentation matrix $\hat{\mathbf{S}}$.

(2) At the q -th we estimate $\hat{\mathbf{Y}}_q$ and $\hat{\mathbf{Z}}_q$ from $r_q(k)$ using the computed segmentation matrix $\hat{\mathbf{S}}$. Next, we apply our parameter estimation algorithm on $\hat{\mathbf{Y}}_q$ which produces the estimates $\hat{\mathbf{F}}_q$, $\hat{\mathbf{K}}_q$, and $\hat{\mathbf{C}}_q$. The q -th sensor transmits $\hat{\mathbf{F}}_q$, $\hat{\mathbf{K}}_q$, and $\hat{\mathbf{C}}_q$ to all other sensors, which can be done using one broadcast message. This process is repeated for $q = 1, \dots, Q$. At each sensor we perform the fusion of $\hat{\mathbf{C}}_q$ for $q = 1, \dots, Q$ which produces the final estimates of activity sequences $\hat{\mathbf{C}}$.

(3) At the q -th sensor we estimate $\hat{\mathbf{G}}_q = [\hat{\mathbf{g}}_{q1} \ \dots \ \hat{\mathbf{g}}_{qM}]$ from $\hat{\mathbf{Z}}_q$ and $\hat{\mathbf{C}}$. The q -th sensor transmits $\hat{\mathbf{G}}_q$ to all other sensors, which can be done using one broadcast message. This process is repeated for $q = 1, \dots, Q$. At each sensor, from the vectors $\hat{\mathbf{g}}_{qm}$ for $q = 1, \dots, Q$ we estimate power spectrum of the m -th transmitter $S_m(f)$ and the set of channel transfer functions $|H_{qm}(f)|^2$ for $q = 1, \dots, Q$. The algorithm for computing these estimates will be presented in the next section. The estimation procedure is repeated for $m = 1, \dots, M$ which is the last step of our algorithm.

4.2 SIMO blind deconvolution

In this section we consider the scenario with one transmitter and Q sensors. The received signals at the sensors are

$$r_q(t) = \int_0^\infty h_{cq}(\tau) s(t - \tau) d\tau \quad (4.1)$$

where $s(t)$ is the transmitted signal and $h_{cq}(t)$ is impulse response of the channel between the q -th sensor and the transmitter for $q = 1, \dots, Q$. The received signal $r_q(t)$ is sampled with the period T_s . Assuming that $s(t)$ is bandlimited to $[-W/2, W/2]$ it can be represented as [100, pp. 150]

$$s(t) = \sum_{n=-\infty}^{\infty} s(nT_s) g_r(t - nT_s) \quad (4.2)$$

where $T_s = 1/W$ and $g(t) = T_s \frac{\sin \pi t/T_s}{\pi t}$. By inserting (4.2) into (4.1) we get the discrete time model

$$r_q(k) = \sum_{n=-\infty}^{\infty} s(n) h_q(k - n) \quad (4.3)$$

where $r_q(k) = r_q(kT_s)$, $s(n) = s(nT_s)$, and $h_q(n) = \int_0^\infty h_{cq}(\tau) g_r(nT_s - \tau) d\tau$. We see that $h_q(n)$ is, in general, nonzero for $-\infty \leq n \leq \infty$ and therefore, it is not a meaningful model for any real world channel. This unrealistic discrete time impulse response $h_q(n)$ resulted from assuming that $s(t)$ is perfectly bandlimited to $[-W/2, W/2]$. In practice, $s(t)$ is only approximately bandlimited to $[-W/2, W/2]$. Let us try to derive a more realistic discrete time model. From (4.1) we get

$$r_q(kT_s) = \sum_{n=0}^{\infty} \int_{nT_s - T_s/2}^{nT_s + T_s/2} h_{cq}(\tau) s(kT_s - \tau) d\tau \quad (4.4)$$

where we assumed that $h_{cq}(t) = 0$ for $t < 0$. Assuming that $s(t) = s(kT_s)$ for $kT_s - T_s/2 < t \leq kT_s + T_s/2$ we obtain

$$r_q(k) = \sum_{n=0}^{\infty} h_q(n) s(k - n) \quad (4.5)$$

where $h_q(n) = \int_{nT_s - T_s/2}^{nT_s + T_s/2} h_{cq}(\tau) d\tau$ which can be expressed as $h_q(n) = \int_0^\infty h_{cq}(\tau) g_r(nT_s - \tau) d\tau$ where $g_r(t) = 1$ for $-T_s/2 < t \leq T_s/2$ and $g_r(t) = 0$ otherwise. Now we have obtained a causal discrete time impulse response. However, we note that the equality

in (4.5) holds only approximately where the approximation becomes more accurate as T_s gets smaller. In contrast to this, the equality in (4.3) holds exactly. If we assume that

$$h_{cq}(t) = \sum_{n=0}^{L_c} a_{q,n} \delta(t - \tau_{q,n}), \quad (4.6)$$

which is a reasonable model for many real world channels [97, ch. 5], we get from (4.5)

$$r_q(k) = \sum_{n=0}^L h_q(n) s(k-n) \quad (4.7)$$

where $LT_s > \tau_{q,L_c}$ for $q = 1, \dots, Q$. From (4.7) we obtain

$$R_{2q}(k, l) = \sum_{n=0}^L \sum_{m=0}^L h_q(n) h_q(m)^* R_{2s}(k-m, l-n+m) \quad (4.8)$$

where $R_{2q}(k, l) = E[r_q(k+l)r_q(k)^*]$ and $R_{2s}(k, l) = E[s(k+l)s(k)^*]$. If $s(k)$ is stationary (i. e., $R_{2s}(k, l) = R_{2s}(l)$), then it follows from (4.8) that $R_{2q}(k, l) = R_{2q}(l)$. In this case we obtain from (4.8)

$$R_{2q}(l) = \sum_{p=-L}^L R_{2hq}(p) R_{2s}(l-p) \quad (4.9)$$

where $R_{2hq}(p) = \sum_{n=0}^{L-p} h_q(n+p)h_q(n)^*$ for $p \geq 0$ and $R_{2hq}(p) = R_{2hq}(-p)^*$ for $p < 0$. If $s(k)$ is cyclostationary with the period N_c (i. e., $R_{2s}(k+N_c, l) = R_{2s}(k, l)$), then it follows from (4.8) that $R_{2q}(k+N_c, l) = R_{2q}(k, l)$. Now, the equation in (4.8) still holds if $R_{2q}(l)$ is replaced with $R_{2q}^{(0)}(l) = \frac{1}{N_c} \sum_{k=0}^{N_c-1} R_{2q}(k, l)$ and $R_{2s}(l)$ is replaced with $R_{2s}^{(0)}(l) = \frac{1}{N_c} \sum_{k=0}^{N_c-1} R_{2s}(k, l)$.

Our deconvolution problem consists in recovering the sequences $R_{2s}(k)$ and $R_{2hq}(k)$ for $q = 1, \dots, Q$ from the sequences observed at the sensors $R_{2q}(k)$ for $q = 1, \dots, Q$. In order to understand why this problem can be solved we apply Z-transform on (4.9) which yields

$$S_{2q}(z) = H_q(z) H_q(z^{-*})^* S_{2s}(z) \quad (4.10)$$

where $S_{2q}(z) = \sum_{k=-\infty}^{\infty} R_{2q}(k) z^{-k}$, $H_q(z) = \sum_{k=0}^L h_q(k) z^{-k}$, and $S_{2s}(z) = \sum_{k=-\infty}^{\infty} R_{2s}(k) z^{-k}$.

We see from (4.10) that each $S_{2q}(z)$ is a product of two terms: $S_{2s}(z)$, which is common for all sensors, and $H_q(z) H_q(z^{-*})^*$, which is different for each sensor. Therefore, if there is no zero common to all the channels $H_q(z)$ for $q = 1, \dots, Q$ it is possible to factor

$S_{2q}(z)$ into $H_q(z)H_q(z^{-*})^*$ and $S_{2s}(z)$ up to a scaling factor. The scaling ambiguity appears because, if we replace $S_{2s}(z)$ with $kS_{2s}(z)$ where $k > 0$ and $H_q(z)$ with $H_q(z)/\sqrt{k}$, there is no change in $S_{2q}(z)$ in (4.10). If all $H_q(z)$ for $q = 1, \dots, Q$ contain the common zero z_0 then the common factor for all $S_{2q}(z)$ becomes $(1 - z_0z^{-1})(1 - z_0^*z)S_{2s}(z)$ and hence, in this case it is not possible to recover correctly $H_q(z)H_q(z^{-*})^*$ and $S_{2s}(z)$ from $S_{2q}(z)$. We observe that the proposed method requires at least two sensors (i. e., $Q \geq 2$). For $Q = 1$ it is not possible to recover $H_1(z)H_1(z^{-*})^*$ and $S_{2s}(z)$ from $S_{21}(z)$ without imposing additional assumptions on $H_1(z)$ or $S_{2s}(z)$.

This type of single input multiple output (SIMO) deconvolution problem appears in a completely different application in [101] and [102] where identifiability conditions were studied. The main identifiability condition is that the channel transfer functions have no common zeros, which agrees with our analysis above. An additional identifiability condition is that the source signal has a sufficient number of modes [102], which means that the source signal must have nonzero spectral components over the observed frequency range. In our application the channel terms $H_q(z)H_q(z^{-*})^*$ can be recovered only over the range of frequencies where $S_{2s}(z)$ has nonzero components.

Using the proposed method we can recover $H_q(z)H_q(z^{-*})^*$ for $q = 1, \dots, Q$ and $S_{2s}(z)$. From $H_q(z)H_q(z^{-*})^*$ we can recover $H_q(z)$ if we assume that $H_q(z)$ is a minimum phase transfer function. However, the minimum phase assumption is not necessarily true for wireless channels. Hence, using the proposed method we can correctly recover the magnitude of $H_q(z)$ but, in general, we cannot recover the phase of $H_q(z)$. This limitation appeared because we considered only the power spectra at the sensors $S_{2q}(z)$, which are not sensitive to the phase of $H_q(z)$. In order to recover the phase of $H_q(z)$, in addition to the power spectra, we need to consider second order cyclostationary spectra or fourth order spectra at the sensors because these statistics are sensitive to the phase of $H_q(z)$. We leave this extension of the algorithm for future work.

We need a numerical algorithm for solving the formulated deconvolution problem. In practice the observed quantities $R_{2q}(k)$ or $S_{2q}(z)$ are not available and they must be replaced with their estimates. Let $\hat{R}_{2q}(k)$ for $k = 0, \pm 1, \pm 2, \dots, \pm K_r$ and $q = 1, \dots, Q$ be the available estimates. In order to solve the deconvolution problem we seek the

estimates of $R_{2s}(k)$ and $R_{2hq}(k)$ which minimize the following criterion function

$$C_{ls} = \sum_{q=1}^Q \sum_{k=0}^{K_r} |\hat{R}_{2q}(k) - \sum_{p=-L}^L R_{2hq}(p) R_{2s}(k-p)|^2 \quad (4.11)$$

where we used the fact $R_{2q}(k) = R_{2q}(-k)^*$ and considered only $k \geq 0$. The criterion function does not have a unique minimum point because of the scaling ambiguity mentioned earlier. In order to remove this scaling ambiguity we assume $R_{2s}(0) = 1$, which is equivalent to $\int_{-0.5}^{0.5} S_{2s}(e^{j2\pi f}) df = 1$. Now we wish to minimize $C_{ls}(\mathbf{p})$ where

$$\mathbf{p} = [\mathbf{p}_1^T \quad \dots \quad \mathbf{p}_Q^T \quad \mathbf{p}_s^T]^T \quad (4.12)$$

where

$$\mathbf{p}_q = [R_{2hq}(0) \quad \text{Re}\{R_{2hq}(1)\} \quad \dots \quad \text{Re}\{R_{2hq}(L)\} \quad \text{Im}\{R_{2hq}(1)\} \quad \dots \quad \text{Im}\{R_{2hq}(L)\}]^T \quad (4.13)$$

for $q = 1, \dots, Q$ and

$$\mathbf{p}_s = [\text{Re}\{R_{2s}(1)\} \quad \dots \quad \text{Re}\{R_{2s}(K_r + L)\} \quad \text{Im}\{R_{2s}(1)\} \quad \dots \quad \text{Im}\{R_{2s}(K_r + L)\}]^T. \quad (4.14)$$

While minimizing $C_{ls}(\mathbf{p})$ we have to take into account $\sum_{k=-L}^L R_{2hq}(k) e^{-j2\pi f k} \geq 0$ and $\sum_{k=-(L+K_r)}^{K_r+L} R_{2s}(k) e^{-j2\pi f k} \geq 0$ for $-0.5 \leq f \leq 0.5$. The constraints on $R_{2hq}(k)$ can be taken into account at the finite number of frequencies $f_r = r/N_f$ for $r = 0, \dots, N_f - 1$ and expressed as $\mathbf{A}_h \mathbf{p}_q \geq 0$ for $q = 1, \dots, Q$ where

$$\mathbf{A}_h = \begin{bmatrix} 1 & 2 \cos 2\pi f_1 & \dots & 2 \cos 2\pi f_1 L & 2 \sin 2\pi f_1 & \dots & 2 \sin 2\pi f_1 L \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 2 \cos 2\pi f_{N_f} & \dots & 2 \cos 2\pi f_{N_f} L & 2 \sin 2\pi f_{N_f} & \dots & 2 \sin 2\pi f_{N_f} L \end{bmatrix}. \quad (4.15)$$

Assuming $R_{2s}(0) = 1$ the constraint on $R_{2s}(k)$ can be expressed as $\mathbf{A}_s \mathbf{p}_s + \mathbf{1} \geq 0$ where

$$\mathbf{A}_s = \begin{bmatrix} 2 \cos 2\pi f_1 & \dots & 2 \cos 2\pi f_1 (K_r + L) & 2 \sin 2\pi f_1 & \dots & 2 \sin 2\pi f_1 (K_r + L) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 2 \cos 2\pi f_{N_f} & \dots & 2 \cos 2\pi f_{N_f} (K_r + L) & 2 \sin 2\pi f_{N_f} & \dots & 2 \sin 2\pi f_{N_f} (K_r + L) \end{bmatrix}. \quad (4.16)$$

Thus, our optimization problem is to minimize $C_{ls}(\mathbf{p})$ subject to $\mathbf{A}_h \mathbf{p}_q \geq 0$ for $q = 1, \dots, Q$ and $\mathbf{A}_s \mathbf{p}_s + \mathbf{1} \geq 0$.

When the channels $H_q(z)$ for $q = 1, \dots, Q$ have no common zeros our deconvolution problem has a unique solution which can be found by minimizing $C_{ls}(\mathbf{p})$. The quality of the obtained solution depends on the conditioning of the problem. When $R_{2q}(k)$ is replaced with $\hat{R}_{2q}(k)$ some perturbation of $R_{2q}(k)$ is created. If the problem is well conditioned the obtained solution is robust to the perturbation of $R_{2q}(k)$. On the other hand, if the problem is ill conditioned the obtained solution is very sensitive to the perturbation of $R_{2q}(k)$. It turns out that this type of blind deconvolution problem can be ill conditioned in some practical situations. In order to deal with this problem we consider the following criterion function

$$C_{rls}(\mathbf{p}) = C_{ls}(\mathbf{p}) + \lambda \sum_{q=1}^Q \|\mathbf{p}_q\|_1 \quad (4.17)$$

where the parameter $\lambda \geq 0$ controls the strength of the added regularization term. The solution obtained by minimizing $C_{rls}(\mathbf{p})$ is a compromise between minimizing $C_{ls}(\mathbf{p})$ and minimizing the regularization term. It was observed that this type of l_1 norm regularization term favors sparse solutions. Hence, it is used in problems where among multiple possible solution we wish to select the simplest or sparsest solution [92] [103] [94] [33]. This type of l_1 norm regularization is also used in deconvolution problems [104] [105]. However, in these deconvolution problems the input signal is assumed to be known. Our deconvolution problem is fundamentally different since the input signal $R_{2s}(k)$ is not known and must be estimated along with the channel parameters $R_{2qh}(k)$. Hence, the methods from [104] or [105] cannot be applied to our problem.

Our first step in solving the problem of minimizing $C_{rls}(\mathbf{p})$ is to express

$$\mathbf{p}_q = \mathbf{u}_q - \begin{bmatrix} 0 \\ \mathbf{v}_q \end{bmatrix} \quad (4.18)$$

where $\mathbf{u}_q \geq 0$, $\mathbf{v}_q \geq 0$, $q = 1, \dots, Q$ and we use the fact that $R_{2hq}(0) \geq 0$. This transformation is a standard step in solving problems with l_1 norm regularization [94] [33]. The constraints $\mathbf{A}_h \mathbf{p}_q \geq 0$ can be expressed as

$$\begin{bmatrix} \mathbf{A}_h & -\mathbf{A}_{hc} \end{bmatrix} \begin{bmatrix} \mathbf{u}_q \\ \mathbf{v}_q \end{bmatrix} \geq 0 \quad (4.19)$$

where \mathbf{A}_{hc} contains the last $2L$ columns of \mathbf{A}_h (i. e., $\mathbf{A}_h = \begin{bmatrix} \mathbf{1} & \mathbf{A}_{hc} \end{bmatrix}$) and $q = 1, \dots, Q$.
If we define the parameter vector

$$\mathbf{r} = \begin{bmatrix} \mathbf{u}_1^T & \mathbf{v}_1^T & \dots & \mathbf{u}_Q^T & \mathbf{v}_Q^T & \mathbf{p}_s^T \end{bmatrix}^T \quad (4.20)$$

the criterion function can be expressed as

$$C_{rls}(\mathbf{r}) = C_{ls}(\mathbf{r}) + \lambda \begin{bmatrix} \mathbf{1}^T & \mathbf{0}^T \end{bmatrix} \mathbf{r} \quad (4.21)$$

where $\mathbf{1}$ is a vector of $Q(4L + 1)$ ones and $\mathbf{0}$ is a vector of $2(K_r + L)$ zeros. Our optimization problem is

$$\hat{\mathbf{r}} = \underset{\mathbf{A}_{in}\mathbf{r} + \mathbf{b} \geq 0}{\text{argmin}} C_{rls}(\mathbf{r}) \quad (4.22)$$

where

$$\mathbf{A}_{in} = \begin{bmatrix} \tilde{\mathbf{A}}_h & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{A}}_h & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \tilde{\mathbf{A}}_h & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{A}_s \\ \mathbf{I} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{I} & \mathbf{0} \end{bmatrix}, \quad (4.23)$$

where $\tilde{\mathbf{A}}_h = \begin{bmatrix} \mathbf{A}_h & -\mathbf{A}_{hc} \end{bmatrix}$, \mathbf{I} is a $4L + 1$ by $4L + 1$ identity matrix and

$$\mathbf{b} = \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \\ \mathbf{0} \end{bmatrix} \quad (4.24)$$

where the top $\mathbf{0}$ is a vector of QN_f zeros, $\mathbf{1}$ is a vector of N_f ones and the bottom $\mathbf{0}$ is a vector of $Q(4L + 1)$ zeros.

In order to solve (4.22) we follow the same procedure used in the previous chapter. Our first step is to eliminate the inequality constraints using the logarithmic barrier function method. We define $\mathbf{c} = \begin{bmatrix} c_1 & \dots & c_{N_{in}} \end{bmatrix}^T = \mathbf{A}_{in}\mathbf{r} + \mathbf{b}$ where $N_{in} = (Q +$

$1)N_f + Q(4L + 1)$ is the total number of inequality constraints and consider the criterion function

$$C_{bf}(\mathbf{r}) = C_{rls}(\mathbf{r}) - \alpha \sum_{i=1}^{N_{in}} \log c_i \quad (4.25)$$

where α is a small positive constant which controls the influence of the added logarithmic terms. When the i -th inequality constraint is about to be violated c_i approaches zero and its logarithmic term tends to negative infinity. Then $C_{bf}(\mathbf{r})$ tends to plus infinity, which prevents that parameter from becoming zero or negative. Now our optimization problem is to minimize $C_{bf}(\mathbf{r})$. Again, we will use Newton's method starting from an initial point for \mathbf{r} . At each iteration we update the parameter vector using

$$\mathbf{r} = \mathbf{r}_0 - \eta \mathbf{H}(\mathbf{r}_0)^{-1} \mathbf{g}(\mathbf{r}_0) \quad (4.26)$$

where \mathbf{r}_0 is the parameter vector from the previous iteration, η is the step size, $\mathbf{g}(\mathbf{r})$ is the gradient and $\mathbf{H}(\mathbf{r})$ is the Hessian of $C_{bf}(\mathbf{r})$. Next, we need to compute $\mathbf{g}(\mathbf{r})$ and $\mathbf{H}(\mathbf{r})$. Let us define the vectors

$$\begin{aligned} \hat{\mathbf{y}} &= \begin{bmatrix} \hat{\mathbf{y}}_1^T & \dots & \hat{\mathbf{y}}_Q^T \end{bmatrix}^T \\ \mathbf{y}(\mathbf{r}) &= \begin{bmatrix} \mathbf{y}_1^T & \dots & \mathbf{y}_Q^T \end{bmatrix}^T \end{aligned} \quad (4.27)$$

where

$$\begin{aligned} \hat{\mathbf{y}}_q &= \begin{bmatrix} \hat{R}_{2q}(0) & \text{Re}\{\hat{R}_{2q}(1)\} & \dots & \text{Re}\{\hat{R}_{2q}(K_r)\} & \text{Im}\{\hat{R}_{2q}(1)\} & \dots & \text{Im}\{\hat{R}_{2q}(K_r)\} \end{bmatrix}^T \\ \mathbf{y}_q(\mathbf{r}) &= \begin{bmatrix} R_{2q}(0) & \text{Re}\{R_{2q}(1)\} & \dots & \text{Re}\{R_{2q}(K_r)\} & \text{Im}\{R_{2q}(1)\} & \dots & \text{Im}\{R_{2q}(K_r)\} \end{bmatrix}^T. \end{aligned} \quad (4.28)$$

In (4.28) we emphasized the functional dependence of the model vector $\mathbf{y}(\mathbf{r})$ on the parameter vector. The functional dependence is given by (4.9). The criterion function can be expressed as

$$C_{bf}(\mathbf{r}) = \sum_{n=1}^{N_{eq}} [\hat{y}_n - y_n(\mathbf{r})]^2 - \alpha \sum_{m=1}^{N_{in}} \log c_m + \lambda \sum_{m=1}^{Q(4L+1)} r_m \quad (4.29)$$

where $N_{eq} = Q(2K_r + 1)$ is the total number of observations, \hat{y}_m and $y_m(\mathbf{r})$ are the m -th entries of $\hat{\mathbf{y}}$ and $\mathbf{y}(\mathbf{r})$ respectively, c_m is the m -th entry of $\mathbf{c} = \mathbf{A}_{in}\mathbf{r} + \mathbf{b}$ and r_m

is the m -th entry of \mathbf{r} . The gradient of $C_{bf}(\mathbf{r})$ is

$$\mathbf{g}(\mathbf{r}) = -2\mathbf{J}(\mathbf{r})^T [\hat{\mathbf{y}} - \mathbf{y}(\mathbf{r})] - \alpha \mathbf{A}_{in}^T \begin{bmatrix} c_1^{-1} \\ \vdots \\ c_{N_{in}}^{-1} \end{bmatrix} + \lambda \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix} \quad (4.30)$$

where $\mathbf{J}(\mathbf{r}) = \left[\frac{\partial y_n(\mathbf{r})}{\partial r_m} \right]$ for $n = 1, \dots, N_{eq}$, $m = 1, \dots, N_{par}$, where $N_{par} = Q(4L + 1) + 2(K_r + L)$, is the Jacobian matrix [55] of $\mathbf{y}(\mathbf{r})$, $\mathbf{1}$ is a vector of $Q(4L + 1)$ ones and $\mathbf{0}$ is a vector of $2(K_r + L)$ zeros. The Hessian of $C_{bf}(\mathbf{r})$ is

$$\mathbf{H}(\mathbf{r}) = 2\mathbf{J}(\mathbf{r})^T \mathbf{J}(\mathbf{r}) - 2 \sum_{n=1}^{N_{eq}} [\hat{y}_n - y_n(\mathbf{r})] \mathbf{G}_n(\mathbf{r}) + \alpha \mathbf{A}_{in}^T \begin{bmatrix} c_1^{-2} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & c_{N_{in}}^{-2} \end{bmatrix} \mathbf{A}_{in} \quad (4.31)$$

where $\mathbf{G}_n(\mathbf{r}) = \left[\frac{\partial^2 y_n(\mathbf{r})}{\partial r_m \partial r_k} \right]$ for $m, k = 1, \dots, N_{par}$ is the Hessian of the function $y_n(\mathbf{r})$. We see from (4.31) that the first and third terms are positive definite matrices while the second term in the sum is in general indefinite. The iterations from (4.26) cannot be used if $\mathbf{H}(\mathbf{r}_0)$ is indefinite because then $\mathbf{H}(\mathbf{r}_0)^{-1} \mathbf{g}(\mathbf{r}_0)$ may not be a descent direction for $C_{bf}(\mathbf{r})$. Again we use a modified Newton's method where the true Hessian $\mathbf{H}(\mathbf{p})$ in (4.26) is replaced with its positive definite part

$$[\mathbf{H}(\mathbf{r})]_{pd} = 2\mathbf{J}(\mathbf{r})^T \mathbf{J}(\mathbf{r}) + \alpha \mathbf{A}_{in}^T \begin{bmatrix} c_1^{-2} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & c_{N_{in}}^{-2} \end{bmatrix} \mathbf{A}_{in}. \quad (4.32)$$

After this modification it is possible to select a sufficiently small step size η such that $C_{bf}(\mathbf{r}) < C_{bf}(\mathbf{r}_0)$. This is possible since $[\mathbf{H}(\mathbf{r}_0)]_{pd}^{-1} \mathbf{g}(\mathbf{r}_0)$ is a descent direction for $C_{bf}(\mathbf{r})$. In order to select the step size we compute the updated parameter vector as

$$\mathbf{r} = \mathbf{r}_0 - \mu^v [\mathbf{H}(\mathbf{r}_0)]_{pd}^{-1} \mathbf{g}(\mathbf{r}_0) \quad (4.33)$$

where $0 < \mu < 1$ and $v = 0, 1, 2, \dots, V_{at}$. We start with $v = 0$ and keep increasing v until all inequality constraints are satisfied and $C_{bf}(\mathbf{r}) < C_{bf}(\mathbf{r}_0)$. The step size selection is the final part of the algorithm.

Let us summarize the algorithm for solving (4.22). Starting from a random initial point, which satisfies all inequality constraints from (4.22), we iteratively update the

parameter vector \mathbf{r} as follows. In each iteration we compute $\mathbf{J}(\mathbf{r})$ as partial derivatives of model equations (4.9), $[\mathbf{H}(\mathbf{r})]_{pd}$ using (4.32), $\mathbf{g}(\mathbf{r})$ using (4.30), select the step size using the proposed line search procedure and update the parameter vector using (4.33). The iterations are repeated until convergence. The algorithm converges to a local minimum of the criterion function $C_{bf}(\mathbf{r})$. The global minimum of $C_{bf}(\mathbf{r})$ can be found from the results obtained by running the algorithm from different initial points.

As already mentioned this type of SIMO blind deconvolution problem was studied in [101] and [102] where certain algebraic solutions were proposed. Main problem with these solutions is that they produce estimates which do not minimize any meaningful criterion function. We see from (4.11) that in our application some amount of estimation error is always present. Since the algebraic solutions do not minimize any meaningful criterion function they are not robust in the presence of such modeling errors. Another problem with these algebraic solutions is that they cannot take into account any application specific constraints such as nonnegativity or sparsity. Therefore, if the amount of estimation error is high or the problem is ill conditioned the algebraic solutions are often useless. These facts explain why a new numerical algorithm was needed. The proposed algorithm produces estimates by minimizing a meaningful criterion function and allows incorporation of various application specific constraints.

4.3 Numerical example

We consider the setup with one transmitter four sensors shown in Figure 2.16. The transmitter transmits the same type of signal that was described in section 2.3 and used to study the segmentation algorithm. Channel transfer functions between the transmitter and the sensors are measured in ORBIT room in WINLAB for the setup shown in Figure 2.16 [66]. These channel transfer functions are used in our simulations. Figure 2.17 shows an example of power traces of the received signals at the sensors. Using the steps outlined in section 4.1 at each sensor we can estimate the power spectrum of the received source signal. Figure 4.1 shows the estimated power spectra for $SNR = 10dB$ where $SNR_1 = SNR + 8.42dB$, $SNR_2 = SNR + 5.44dB$, $SNR_3 = SNR$, and $SNR_4 = SNR + 5.44dB$. From these four power spectra we wish to estimate the

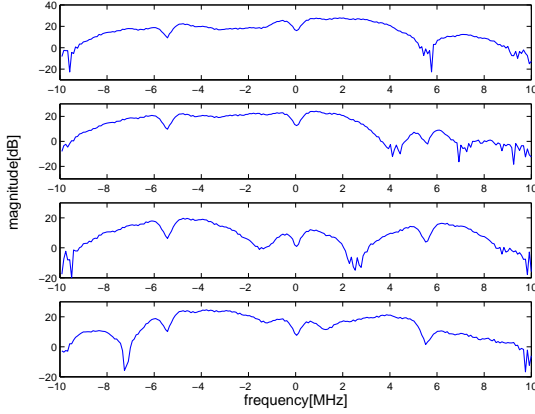


Figure 4.1: Estimated PSDs at four sensors for SNR=10dB

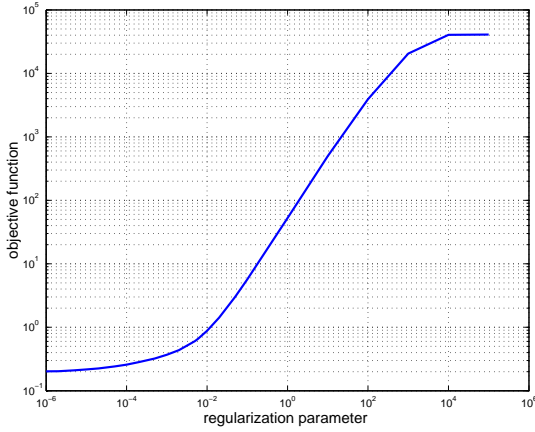


Figure 4.3: Objective function vs. regularization parameter

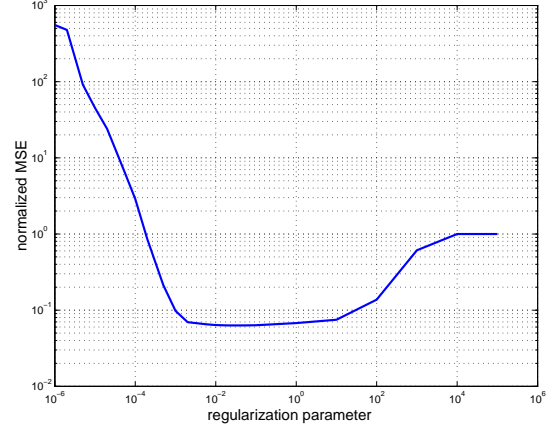


Figure 4.2: Normalized squared estimation error of the channel estimates vs. regularization parameter

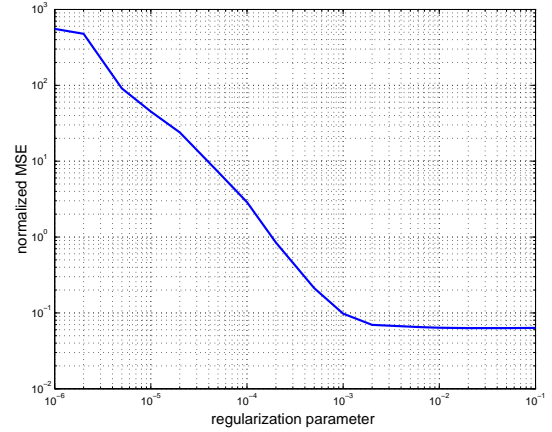


Figure 4.4: Normalized squared estimation error of the channel estimates vs. regularization parameter

source power spectrum and magnitude frequency responses of the channels between the source and the sensors. In order to use the proposed algorithm we need to know the length of the impulse responses L . We examined the impulse responses from [66] and determined that they all become practically zero after $2\mu s$. Since in our simulations $T_s = 0.05\mu s$ we select $L = 39$. We apply our algorithm with $K_r = 120$ and $L = 39$ for different values of the regularization parameter λ . For each value of λ we compute

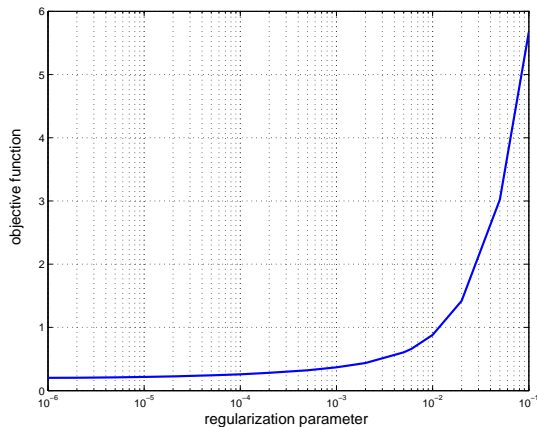


Figure 4.5: Objective function vs. regularization parameter

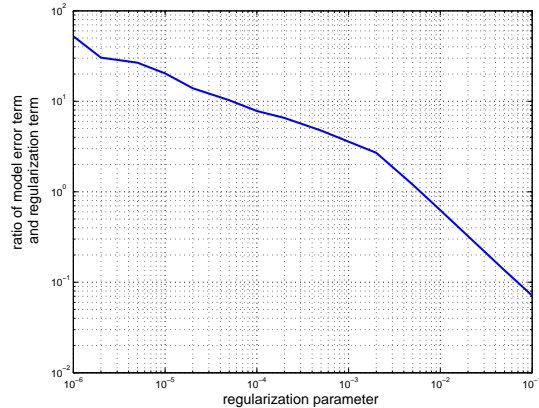


Figure 4.6: Ratio of the squared error term and the regularization term vs. regularization parameter

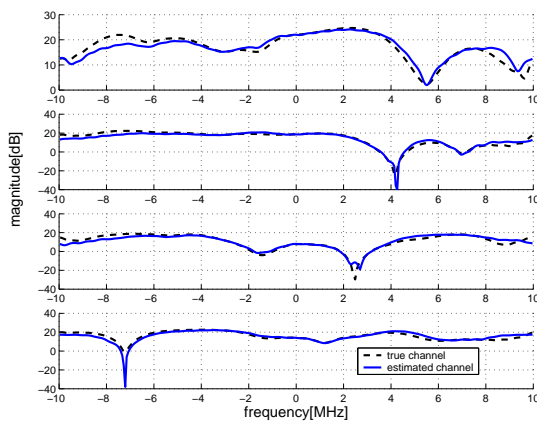


Figure 4.7: True and estimated magnitude frequency responses of the channels

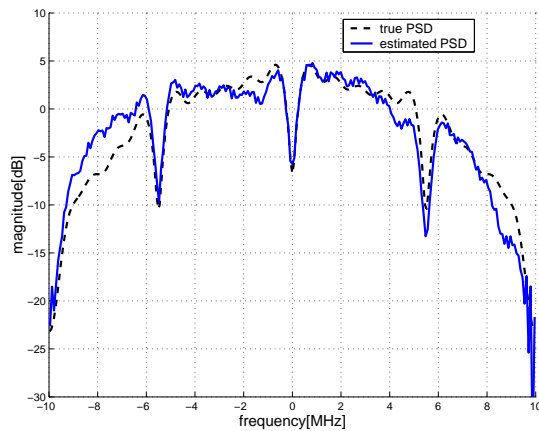


Figure 4.8: True and estimated source PSD for SNR=10 dB and observation time of 5ms

the normalized square estimation error of the estimated channel parameters defined as $\|\mathbf{p}_{ch} - \hat{\mathbf{p}}_{ch}\|_2^2 / \|\mathbf{p}_{ch}\|_2^2$ where $\mathbf{p}_{ch} = [\mathbf{p}_1^T \ \dots \ \mathbf{p}_Q^T]^T$. Figure 4.2 the normalized squared estimation error versus λ . We see that for very small values of λ the estimation error is very high and hence, the obtained estimates are useless. This can be explained by the fact that the problem is extremely ill conditioned. The estimation error present in the power spectra Figure 4.1 and in the corresponding autocorrelation sequences affects the computed solution severely and thus, causes the huge estimation error. As λ

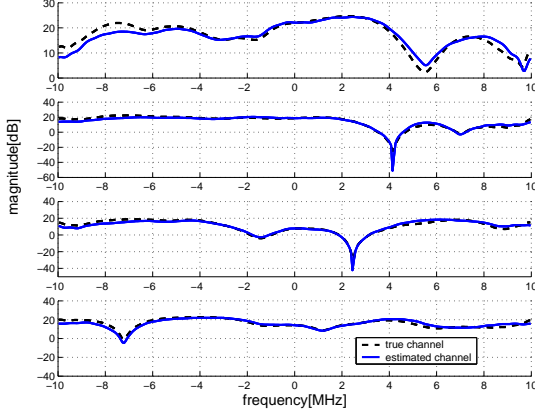


Figure 4.9: True and estimated magnitude frequency responses of the channels

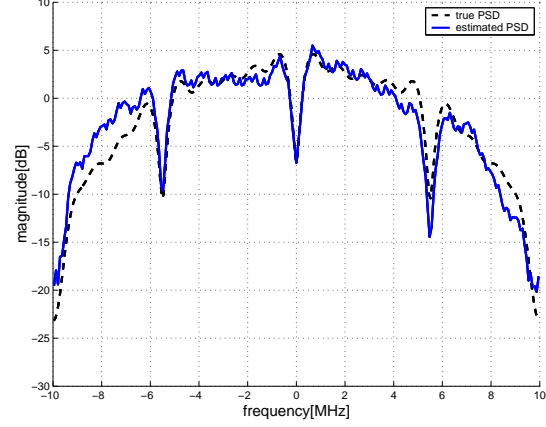


Figure 4.10: True and estimated source PSD for SNR=30 dB and observation time of 5ms

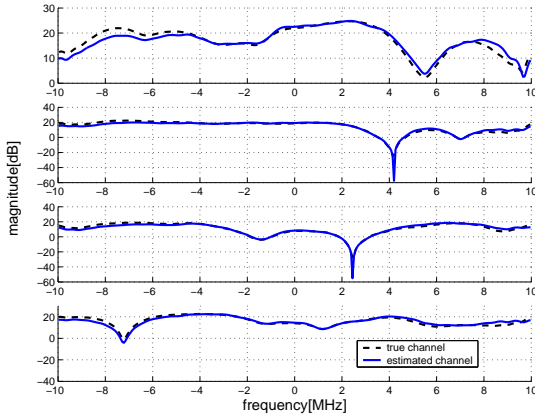


Figure 4.11: True and estimated magnitude frequency responses of the channels

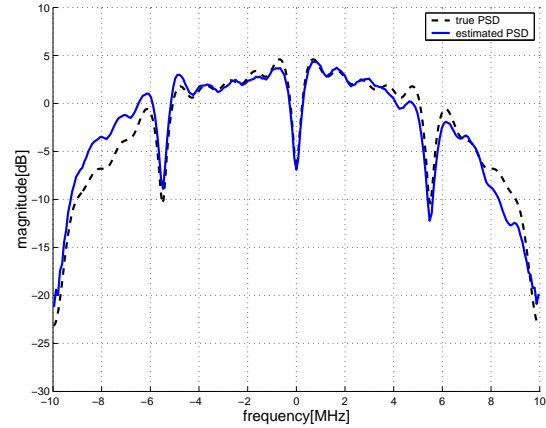


Figure 4.12: True and estimated source PSD for SNR=30 dB and observation time of 50ms

increases the channel parameter estimation error drops and reaches its minimum for a certain value of λ . Further increase of λ increases the estimation error because the regularization term becomes too strong and restricts the estimated parameters too much. When λ is very large the estimated parameters $\hat{\mathbf{p}}_{ch}$ tend to zeros and the normalized estimation error shown in Figure 4.2 tends to one. Figure 4.3 shows $C_{rls}(\hat{\mathbf{r}})$, where the criterion function $C_{rls}(\mathbf{r})$ is defined in (4.21) and $\hat{\mathbf{r}}$ is the estimated parameter vector, for different values of λ . When λ tends to zero $C_{rls}(\hat{\mathbf{r}})$ becomes constant and equal

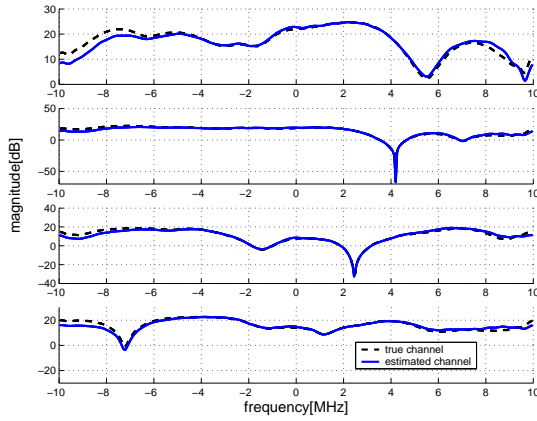


Figure 4.13: True and estimated magnitude frequency responses of the channels

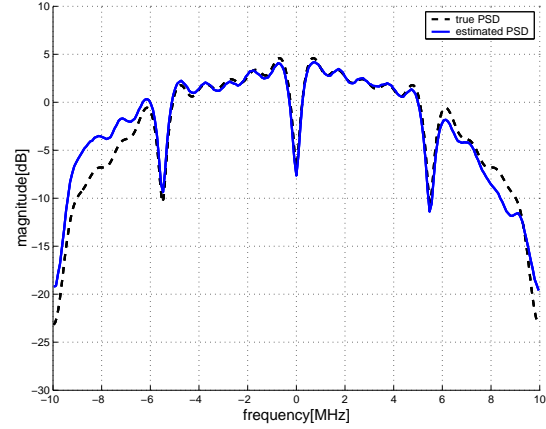


Figure 4.14: True and estimated source PSD for SNR=30 dB and observation time of 250ms

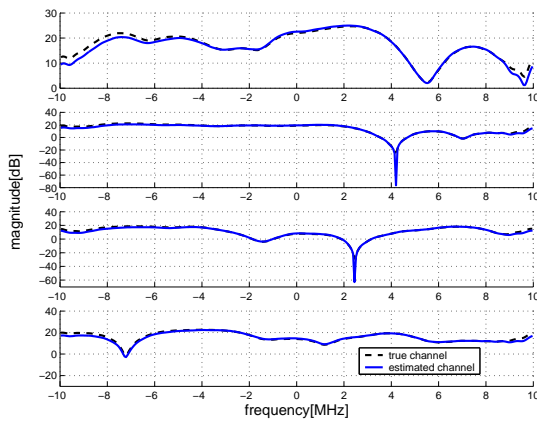


Figure 4.15: True and estimated magnitude frequency responses of the channels

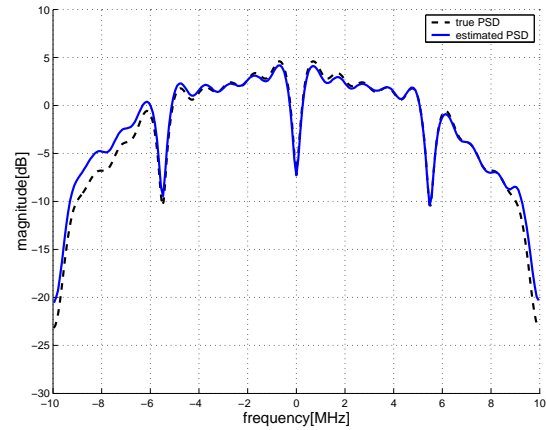


Figure 4.16: True and estimated source PSD for SNR=30 dB and observation time of 1250ms

to $C_{ls}(\hat{\mathbf{r}})$. As λ increases the minimum value of $C_{rls}(\mathbf{r})$ also increases because of the constraints imposed on the estimated parameter vector $\hat{\mathbf{r}}$ and because of the increasing influence of the regularization term. When λ is very large the estimated channel parameters in $\hat{\mathbf{r}}$ tend to zeros and $C_{rls}(\hat{\mathbf{r}})$ becomes approximately constant. From Figure 4.2 and 4.3 we see that we obtain a useful estimate $\hat{\mathbf{r}}$ when λ is sufficiently large and the influence of the regularization term on $C_{rls}(\hat{\mathbf{r}})$ becomes noticeable. This can be seen more clearly from Figure 4.4 and 4.5 where the relevant parts of the curves

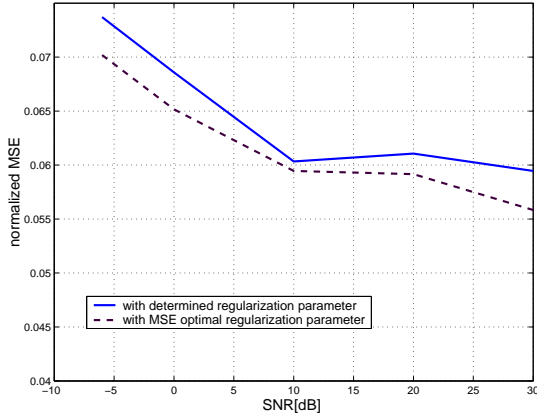


Figure 4.17: Normalized MSE of the channel estimates vs. SNR

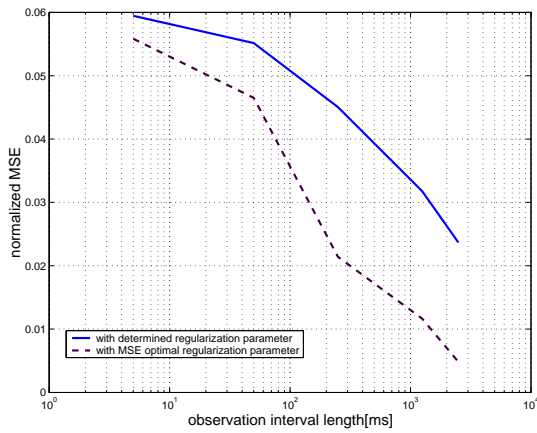


Figure 4.19: Normalized MSE of the channel estimates vs. observation time length

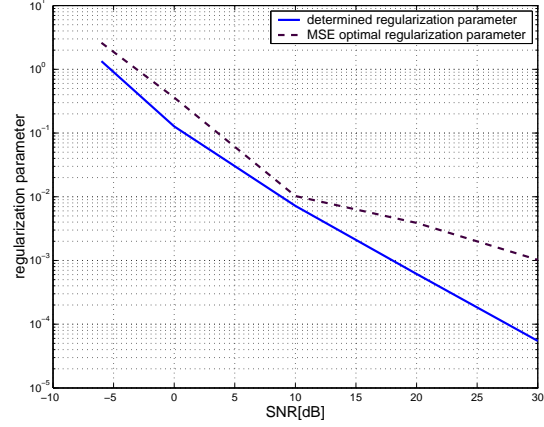


Figure 4.18: Determined and MSE optimal values of the regularization parameter vs. SNR

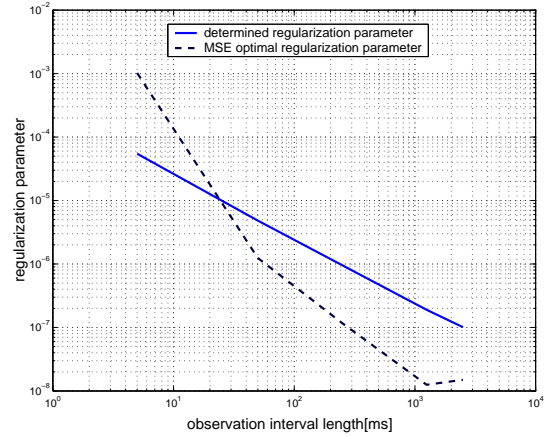


Figure 4.20: Determined and MSE optimal values of the regularization parameter vs. observation time length

from Figure 4.2 and 4.3 are shown. The the criterion function defined in (4.21) can be represented as $C_{rls}(\mathbf{r}) = C_{ls}(\mathbf{r}) + C_{reg}(\mathbf{r})$ where $C_{reg}(\mathbf{r}) = \lambda \begin{bmatrix} \mathbf{1}^T & \mathbf{0}^T \end{bmatrix} \mathbf{r}$. When λ is selected so that $C_{ls}(\hat{\mathbf{r}}) \gg C_{reg}(\hat{\mathbf{r}})$ the influence of the regularization term is negligible and the computed solution $\hat{\mathbf{r}}$ is useless because the problem is ill conditioned. The influence of the regularization becomes noticeable when $C_{reg}(\hat{\mathbf{r}})$ is comparable to $C_{ls}(\hat{\mathbf{r}})$. Therefore, one reasonable strategy is to select λ such that $C_{ls}(\hat{\mathbf{r}}) = C_{reg}(\hat{\mathbf{r}})$. Figure 4.6

shows the ratio $C_{ls}(\hat{\mathbf{r}})/C_{reg}(\hat{\mathbf{r}})$ for different values of λ . We see that this ratio decreases as λ increases. Therefore, we can start with some small λ and keep increasing it until $C_{ls}(\hat{\mathbf{r}}) = C_{reg}(\hat{\mathbf{r}})$. The vector $\hat{\mathbf{r}}$ obtained for λ such that $C_{ls}(\hat{\mathbf{r}}) = C_{reg}(\hat{\mathbf{r}})$ is the output of our algorithm. Figure 4.7 and 4.8 show the estimates obtained for SNR=10 dB and the observation time of 5ms. We see that the estimates contain some estimation error which is caused by the estimation errors in the input autocorrelation sequences $\hat{R}_{2q}(k)$. Figure 4.9 and 4.10 show the estimates obtained for SNR=30 dB and the observation time of 5ms. We do not see a significant improvement because the estimation errors in $\hat{R}_{2q}(k)$ is caused by the finite sample size. This error can be reduced by increasing the observation time which can be seen from the estimates shown in Figure 4.11 through 4.16. Next, we study the behavior of the algorithm for a range of SNR values. At each SNR point we perform 10 runs where in each run we generate independent realizations of the signal and noise and apply our algorithm. For each SNR point we compute the mean normalized mean squared error of the channel estimates defined as $\sum_{i=1}^{N_{at}} \|\mathbf{p}_{ch} - \hat{\mathbf{p}}_{ch}^{(i)}\|_2^2 / \|\mathbf{p}_{ch}\|_2^2$ where $\hat{\mathbf{p}}_{ch}^{(i)}$ is the channel estimate obtained in the i -th run and N_{at} is the total number of runs. This error is computed first for the channel estimates obtained for λ selected so that $C_{ls}(\hat{\mathbf{r}}) = C_{reg}(\hat{\mathbf{r}})$ and then for the channel estimates obtained for λ which minimizes $\|\mathbf{p}_{ch} - \hat{\mathbf{p}}_{ch}^{(i)}\|_2$ in each run. In our simulations the value of λ which minimizes $\|\mathbf{p}_{ch} - \hat{\mathbf{p}}_{ch}^{(i)}\|_2$ is possible to find since \mathbf{p}_{ch} is known. In real world applications \mathbf{p}_{ch} is not known and hence, this method for selection of λ cannot be used. Figure 4.17 shows normalized mean squared error of the channel estimates versus SNR for the fixed observation time of 5ms. We see that the curve obtained for the values of λ computed using our approach is relatively close to the curve obtained for the optimal values of λ . As SNR increases the estimation error reduces, as expected. At very high SNR values the estimation error does not tend to zero because of the estimation errors in the input autocorrelation sequences $\hat{R}_{2q}(k)$ which are caused by the finite sample size. Figure 4.18 shows the values of λ computed using our approach and the optimal values of λ versus SNR. We see that the two curves are relatively close and have similar decreasing trends. Figure 4.19 shows normalized mean squared error of the channel estimates versus the observation

time length for $SNR = 10dB$. As observation time length increases the estimation errors in the input autocorrelation sequences $\hat{R}_{2q}(k)$ decrease which in turn reduces the error of the estimated parameters shown in Figure 4.19. Figure 4.20 shows the values of λ computed using our approach and the optimal values of λ versus observation time length. Again, the two curves exhibit similar decreasing trends. As observation time length increases the estimation errors in $\hat{R}_{2q}(k)$ decrease and thus, the perturbation of the estimated parameters gets smaller. Hence, the value of λ needed to obtain useful parameter estimates decreases as observation time length increases.

We see that the magnitude frequency responses of the channels in this example are complicated functions of frequency, which is always the case in a rich multipath environment. In this type of environment each transmitter location has with a high probability its distinct set of channel transfer functions. Therefore, the recovered channel transfer functions uniquely identify the transmitter location. However, the mapping between the transmitter location and the corresponding channel transfer functions is very complicated and unknown in practical applications. Hence, it is very difficult to estimate the transmitter location from the recovered channel transfer functions.

Recovered transmitted signal spectrum provides information about the type of the used signal and hence, it can be used for signal identification. Similarly, fourth and higher order spectra can also be used for signal identification. Since the received signals spectra in this example are severely distorted by the frequency selective channels it would be difficult to use them for signal identification. Instead, it is much better to use the recovered transmitted signal spectrum obtained after the deconvolution. If multiple sensors are available this technique can be applied in systems for automatic modulation recognition [106] for combating performance degradation caused by multipath.

We have illustrated our algorithm using one specific example. However, if we use other channel transfer functions from [66] we obtain very similar results. Our main conclusion is that SIMO blind deconvolution problems for real world channels cannot be solved using algebraic methods from [101] and [102]. These algorithms are useful only for very simple channels which can be modeled with a small number of coefficients (say, $L < 5$). Even then the algebraic methods are not robust to any perturbation

caused by noise or modeling errors which are always present in practical applications. Real world channels often require much larger number of coefficients (e. g., $L = 39$) and the resulting problems are often ill conditioned. In this environment the algebraic solutions are completely useless. In this work we have developed an algorithm which obtains a useful solution to the SIMO blind deconvolution problem by minimizing a least squares criterion function with the l_1 norm regularization.

4.4 Conclusion and future work

In this chapter we proposed a distributed algorithm which uses observations from multiple sensors. Using the segmentation algorithm and the fourth order spectrum based analysis method we can estimate spectra and the associated activity sequences of the individual signals at each sensor. Thus, for each transmitted signal we can estimate its received power spectrum at each sensor. We proposed a blind deconvolution algorithm which from the received power spectra estimates channel transfer functions between each sensor and the transmitter and the transmitted signal power spectrum. From the estimated channel transfer functions we can estimate the transmitter location. The problem of estimation of the transmitter location from the recovered channel transfer functions is an interesting topic for future work.

Let us now summarize the work done in this thesis. We proposed certain algorithms for analysis of packet based radio signals. Using these algorithms for each transmitted we can estimate its second and fourth order spectrum, its activity sequence in time and the location of its transmitter in space. From these estimated quantities we can form a comprehensive picture of radio spectrum usage in space, time and frequency. As already explained, the obtained information on spectrum usage can be used in some important practical applications.

References

- [1] D. Cabric, “Cognitive radios: System design perspective,” Ph.D. dissertation, Univ. of California, Berkeley, Nov 2007.
- [2] D. Cabric, I. O’Donnell, M.-W. Chen, and R. Brodersen, “Spectrum sharing radios,” *Circuits and Systems Magazine, IEEE*, vol. 6, no. 2, pp. 30–45, 2006.
- [3] C. Cordeiro, K. Challapali, D. Birru, and S. S. Nanandagopalan, “IEEE 802.22: An Introduction to the First Wireless Standard based on Cognitive Radios,” *Journal of Communications*, vol. 1, no. 1, 2006.
- [4] S. M. Kay, *Fundamental of Statistical Signal Processing: Detection Theory*. Upper Saddle River, NJ: Prentice-Hall, 1998.
- [5] H.-S. Chen, W. Gao, and D. Daut, “Spectrum sensing for wireless microphone signals,” *5th IEEE Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops, 2008.*, pp. 1 –5, June 2008.
- [6] Ryan Winfield Woodings and Mark Gerrior, “Avoiding Interference in the 2.4-GHz ISM Band,” *Microwave Engineering Online*, Feb 2005. [Online]. Available: <http://www.k8dac.com/projectnights/cantenna/AvoidingInterference.pdf>
- [7] S. Haykin, D. Thomson, and J. Reed, “Spectrum sensing for cognitive radio,” *Proceedings of the IEEE*, vol. 97, no. 5, pp. 849 –877, may 2009.
- [8] C. M. Spooner, “Multi-resolution white-space detection for cognitive radio,” *MIL-COM 2007.*, pp. 1–9, 29-31 Oct. 2007.
- [9] D. Nogu  t, et al., “Sensing techniques for cognitive radio-state of the art and trends,” *White paper*, April 2009. [Online]. Available: <http://www.scc41.org/>
- [10] W. A. Gardner, *Statistical Spectral Analysis: A Nonprobabilistic Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1987.
- [11] W. A. Gradner, “Signal interception: Unifying theoretical framework for signal detection,” *IEEE Transactions on Communications*, vol. 36, no. 8, pp. 897–906, aug 1998.
- [12] H.-S. Chen, W. Gao, and D. Daut, “Spectrum Sensing Using Cyclostationary Properties and Application to IEEE 802.22 WRAN,” in *IEEE Global Telecommunications Conference, 2007.*, nov. 2007, pp. 3133 –3138.
- [13] H. Rahul, N. Kushman, D. Katabi, C. Sodini, and F. Edalat, “Learning to share: narrowband-friendly wideband networks,” *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 147–158, 2008.

- [14] M. Gandetto and C. Regazzoni, "Spectrum sensing: A distributed approach for cognitive terminals," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 3, pp. 546–557, april 2007.
- [15] J. Nelson, M. Hazen, and M. Gupta, "Global optimization for multiple transmitter localization," *Military Communications Conference, 2006. MILCOM 2006*, pp. 1–7, 23-25 Oct. 2006.
- [16] A. Cichocki and S. Amari, *Adaptive Blind Signal and Image Processing*. West Sussex, UK: John Wiley, 2002.
- [17] J.-F. Cardoso, "Blind signal separation: statistical principles," *Proceedings of the IEEE*, vol. 86, no. 10, pp. 2009–2025, Oct 1998.
- [18] A. Hyvarinen, "Fast and robust fixed-point algorithms for independent component analysis," *Neural Networks, IEEE Transactions on*, vol. 10, no. 3, pp. 626–634, May 1999.
- [19] J. Cardoso and A. Souloumiac, "Blind beamforming for non gaussian signals," *IEE Proceedings(Radar Signal Processing)*, vol. 140, pp. 362–370, dec 1993.
- [20] P. Comon, "Independent component analysis, A new concept?" *Signal Processing*, vol. 36, no. 3, pp. 287–314, Apr 1994.
- [21] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, "A blind source separation technique using second-order statistics," *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, vol. 45, no. 2, pp. 434–444, Feb 1997.
- [22] K. Abed-Meraim, Y. Xiang, J. Manton, and Y. Hua, "Blind source-separation using second-order cyclostationary statistics," *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, vol. 49, no. 4, pp. 694–701, Apr 2001.
- [23] A. Belouchrani and M. Amin, "Blind source separation based on time-frequency signal representations," *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, vol. 46, no. 11, pp. 2888–2897, Nov 1998.
- [24] A. Hyvarinen, "Blind source separation by nonstationarity of variance: a cumulant-based approach," *Neural Networks, IEEE Transactions on*, vol. 12, no. 6, pp. 1471–1474, Nov 2001.
- [25] D.-T. Pham and J.-F. Cardoso, "Blind separation of instantaneous mixtures of nonstationary sources," *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, vol. 49, no. 9, pp. 1837–1848, Sep 2001.
- [26] L. Parra and C. Spence, "Convolutive blind separation of non-stationary sources," *Speech and Audio Processing, IEEE Transactions on*, vol. 8, no. 3, pp. 320–327, May 2000.

- [27] A.-J. van der Veen and A. Paulraj, "An analytical constant modulus algorithm," *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, vol. 44, no. 5, pp. 1136–1155, May 1996.
- [28] S. Talwar, M. Viberg, and A. Paulraj, "Blind separation of synchronous co-channel digital signals using an antenna array. Part I: Algorithms," *IEEE Transactions on Signal Processing*, vol. 44, no. 5, pp. 1184–1197, May 1996.
- [29] A. J. van der Veen, "An analytical method for blind binary signal separation," *IEEE Transactions on Signal Processing*, vol. 45, no. 4, pp. 1078–1082, april 1997.
- [30] A.-J. Van Der Veen, "Algebraic methods for deterministic blind beamforming," *Proceedings of the IEEE*, vol. 86, no. 10, pp. 1987–2008, Oct 1998.
- [31] A. Paulraj and C. Papadias, "Space-time processing for wireless communications," *Signal Processing Magazine, IEEE*, vol. 14, no. 6, pp. 49–83, Nov 1997.
- [32] P. Georgiev, F. Theis, and A. Cichocki, "Sparse component analysis and blind source separation of underdetermined mixtures," *IEEE Transactions on Neural Networks*, vol. 16, no. 4, pp. 992 –996, july 2005.
- [33] Y. Li, A. Cichocki, and S.-i. Amari, "Analysis of Sparse Representation and Blind Source Separation," *Neural Comp.*, vol. 16, no. 6, pp. 1193–1234, 2004.
- [34] P. Stoica and R. Moses, *Introduction to Spectral Analysis*. Upper Saddle River, NJ: Prentice-Hall, 1997.
- [35] D. J. Thomson, "Spectrum estimation and harmonic analysis," *IEEE Proceedings*, vol. 70, no. 9, pp. 1055–1096, sep 1982.
- [36] C. L. Nikias and A. P. Petropulu, *Higher-order Spectra Analysis: A Nonlinear Signal Processing Framework*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
- [37] W. Gardner and C. Spooner, "The cumulant theory of cyclostationary time-series. I. Foundation ," *IEEE Transactions on Signal Processing*, vol. 42, no. 12, pp. 3387 –3408, dec 1994.
- [38] C. Spooner and W. Gardner, "The cumulant theory of cyclostationary time-series. II. Development and applications," *IEEE Transactions on Signal Processing*, vol. 42, no. 12, pp. 3409 –3429, dec 1994.
- [39] W. Gardner, "On the spectral coherence of nonstationary processes," *IEEE Transactions on Signal Processing*, vol. 39, no. 2, pp. 424 –430, feb 1991.
- [40] J. Antoni, "The spectral kurtosis: a useful tool for characterising non-stationary signals," *Mechanical Systems and Signal Processing*, vol. 20, no. 2, pp. 282 – 307, 2006.
- [41] M. Basseville and I. V. Nikiforov, *Detection of abrupt changes: theory and application*. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.
- [42] V. Moskvina and A. Zhigljavsky, "An algorithm based on singular spectrum analysis for change-point detection," *Communications in Statistics-Simulation and Computation*, vol. 32, no. 2, pp. 319–352, 2003.

- [43] B. E Brodsky, B. S Darkhovsky, A. Ya Kaplan, and S. L Shishkin, “A nonparametric method for the segmentation of the eeg,” *Computer methods and programs in biomedicine*, vol. 60, no. 2, pp. 93–106, 1999.
- [44] M. Davy and S. Godsill, “Detection of abrupt spectral changes using support vector machines an application to audio signal segmentation,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2002*, vol. 2, 2002, pp. 1313–1316.
- [45] F. Desobry, M. Davy, and C. Doncarli, “An online kernel change detection algorithm,” *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2961–2974, 2005.
- [46] Y. Kawahara and M. Sugiyama, “Sequential change-point detection based on direct density-ratio estimation,” *Statistical Analysis and Data Mining*, vol. 5, no. 2, pp. 114–127, 2011.
- [47] D. W. Scott, *Multivariate Density Estimation: Theory, Practice and Visualization*. John Wiley and Sons, 1992.
- [48] R. A. Horn and C. R. Johnson, *Matrix analysis*. New York, NY, USA: Cambridge University Press, 1986.
- [49] C. Fraley and A. Raftery, “Model-Based Clustering, Discriminant Analysis, and Density Estimation,” *Journal of the American Statistical Association*, vol. 97, pp. 611–631(21), June 2002.
- [50] K. Fukunaga and L. Hostetler, “The estimation of the gradient of a density function, with applications in pattern recognition,” *IEEE Transactions on Information Theory*, vol. 21, no. 1, pp. 32 – 40, jan 1975.
- [51] Y. Cheng, “Mean shift, mode seeking, and clustering,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 790–799, 1995.
- [52] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, 2002.
- [53] D. Comaniciu, “An algorithm for data-driven bandwidth selection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 281–288, 2003.
- [54] X. Li, Z. Hu, and F. Wu, “A note on the convergence of the mean shift,” *Pattern Recognition*, vol. 40, no. 6, pp. 1756 – 1762, 2007.
- [55] P. E. Gill, W. Murray, and M. H. Wright, *Practical optimization*. London: Academic Press, 1981, 1981.
- [56] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2004.
- [57] D. Comaniciu, V. Ramesh, and P. Meer, “The variable bandwidth mean shift and data-driven scale selection,” in *Proceedings of the Eighth IEEE International Conference on Computer Vision, 2001*, vol. 1, 2001, pp. 438 –445.

- [58] K. Fukunaga, *Introduction to statistical pattern recognition (2nd ed.)*. San Diego, CA, USA: Academic Press Professional, Inc., 1990.
- [59] B. W. Silverman, "Using kernel density estimates to investigate multimodality," *Journal of the Royal Statistical Society, Ser. B*, vol. 43, pp. 97–99, 1981.
- [60] D. W. Muller and G. Sawitzki, "Excess mass estimates and tests for multimodality," *Journal of the American Statistical Association*, vol. 86, no. 415, pp. pp. 738–746, 1991.
- [61] N. I. Fisher and J. S. Marron, "Mode testing via the excess mass estimate," *Biometrika*, vol. 88, no. 2, pp. pp. 499–517, 2001.
- [62] M. C. Minnotte, "Nonparametric testing of the existence of modes," *The Annals of Statistics*, vol. 25, no. 4, pp. pp. 1646–1660, 1997.
- [63] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, april 1979.
- [64] J. Astola, P. Haavisto, and Y. Neuvo, "Vector median filters," *Proceedings of the IEEE*, vol. 78, no. 4, pp. 678–689, apr 1990.
- [65] D. Comaniciu and P. Meer, "Distribution free decomposition of multivariate data," *Pattern Analysis and Applications*, vol. 2, pp. 22–30, 1999.
- [66] H. Kremo, J. Lei, I. Seskar, L. Greenstein, and P. Spasojevic, "Characterization of the orbit indoor testbed radio environment," *VTC-2007 Fall*, pp. 946–950, Oct 2007.
- [67] "Wireless LAN Medium Access Control and Physical Layer Specifications," ANSI/IEEE, 1999.
- [68] D. R. Brillinger, "An introduction to polyspectra," *The Annals of Mathematical Statistics*, vol. 36, no. 5, pp. 1351–1374, 1965.
- [69] D. J. Thomson, "Multitaper analysis of nonstationary and nonlinear time series," in *Nonlinear and nonstationary signal processing*, W. J. Fitzgerald, R. L. Smith, A. T. Walden, and P. C. Young, Eds. Cambridge, UK: Cambridge University Press, 2000, ch. 11, pp. 317–394.
- [70] P. O. Amblard, M. Gaeta, and J. L. Lacoume, "Statistics for complex variables and signals—Part II: signals," *Signal Process.*, vol. 53, no. 1, pp. 15–25, 1996.
- [71] D. Ruppert, "What is kurtosis?: An influence function approach," *The American Statistician*, vol. 41, no. 1, pp. pp. 1–5, 1987.
- [72] A. J. van der Veen, "Algebraic constant modulus algorithm," in *Signal Processing Advances in Wireless and Mobile Communications*, G. B. Giannakis, Y. Hua, P. Stoica, and L. Tong, Eds. Upper Saddle River, NJ: Prentice Hall, 2001, ch. 3, pp. 89–130.

- [73] B. Afsari, "Sensitivity analysis for the problem of matrix joint diagonalization," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1148–1171, 2008.
- [74] A. Yeredor, "Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation," *IEEE Transactions on Signal Processing*, vol. 50, no. 7, pp. 1545–1553, Jul 2002.
- [75] J. Kruskal, "Rank decomposition and uniqueness for 3-way and n-way arrays," in *Multiway Data Analysis*, R. Coppi and S. Bolasco, Eds. Amsterdam: Elsevier North Holland, 1988, pp. 8–18.
- [76] N. Sidiropoulos and R. Bro, "Parafac techniques for signal separation," in *Signal Processing Advances in Wireless and Mobile Communications*, G. B. Giannakis, Y. Hua, P. Stoica, and L. Tong, Eds. Upper Saddle River, NJ: Prentice Hall, 2001, ch. 4, pp. 131–179.
- [77] J. B. Kruskal, "Three-way arrays: Rank and uniqueness of trilinear decomposition with applications to arithmetic complexity and statistics," *Linear algebra and its applications*, vol. 18, pp. 95–138, 1977.
- [78] H. Ghennioui, F. El Mostafa, N. Thirion-Moreau, A. Adib, and E. Moreau, "A nonunitary joint block diagonalization algorithm for blind separation of convolutive mixtures of sources," *IEEE Signal Processing Letters*, vol. 14, no. 11, pp. 860–863, nov. 2007.
- [79] A. Belouchrani, M. Amin, and K. Abed-Meraim, "Direction finding in correlated noise fields based on joint block-diagonalization of spatio-temporal correlation matrices," *IEEE Signal Processing Letters*, vol. 4, no. 9, pp. 266–268, sep 1997.
- [80] L. D. Lathauwer, "Decompositions of a Higher-Order Tensor in Block Terms—Part II: Definitions and Uniqueness," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1033–1066, 2008.
- [81] J. F. Cardoso, "Multidimensional independent component analysis," in *International conference on acoustics, speech and signal processing*, may 1998, pp. 1941–1944.
- [82] P. Bradley and O. Mangasarian, "k-Plane Clustering," *Journal of Global Optimization*, vol. 16, no. 1, pp. 23–32, Jan. 2000.
- [83] C. F. V. Loan, "The ubiquitous kronecker product," *Journal of Computational and Applied Mathematics*, vol. 123, no. 1-2, pp. 85–100, Nov 2000.
- [84] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, September 2009.
- [85] P. Huber, B. Kleiner, T. Gasser, and G. Dumermuth, "Statistical methods for investigating phase relations in stationary stochastic processes," *IEEE Transactions on Audio and Electroacoustics*, vol. 19, no. 1, pp. 78–86, mar 1971.
- [86] C. Nikias and J. Mendel, "Signal processing with higher-order spectra," *IEEE Signal Processing Magazine*, vol. 10, no. 3, pp. 10–37, jul 1993.

- [87] J. R. Barry, E. A. Lee, and D. G. Messerschmitt, *Digital Communication*. Norwell, Massachusetts, USA: Kluwer Academic Publishers, 2004.
- [88] R. Bro, “Multi-way analysis in the food industry: Models, algorithms, and applications,” Ph.D. dissertation, The Royal Veterinary and Agricultural University, Denmark, Nov 1998.
- [89] P. Paatero, “A weighted non-negative least squares algorithm for three-way PARAFAC factor analysis,” *Chemometrics and Intelligent Laboratory Systems*, vol. 38, October 1997.
- [90] —, “The Multilinear Engine: A Table-Driven, Least Squares Program for Solving Multilinear Problems, including the n-Way Parallel Factor Analysis Model,” *Journal of Computational and Graphical Statistics*, vol. 8, no. 4, pp. 854–888, Dec., 1999.
- [91] M. Wax and T. Kailath, “Detection of signals by information theoretic criteria,” *IEEE Transactions on Signal Processing*, vol. ASSP-33, no. 2, pp. 387–392, april 1985.
- [92] E. J. Candes, J. K. Romberg, and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.
- [93] D. Donoho, M. Elad, and V. Temlyakov, “Stable recovery of sparse overcomplete representations in the presence of noise,” *IEEE Transactions on Information Theory*, vol. 52, no. 1, pp. 6–18, Jan. 2006.
- [94] S. S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic decomposition by basis pursuit,” *SIAM Journal on Scientific Computing*, vol. 20, pp. 33–61, December 1998.
- [95] C. A. Sugar and G. M. James, “Finding the number of clusters in a dataset: An information-theoretic approach,” *Journal of the American Statistical Association*, vol. 98, pp. 750–763(14), 1 September 2003.
- [96] J. C. Hartsen, “Bluetooth radio system,” *IEEE Personal Communications*, vol. 7, no. 1, pp. 28–36, Feb 2000.
- [97] T. Rappaport, *Wireless Communications: Principles and Practice*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.
- [98] J. Elson and D. Estrin, “Time synchronization for wireless sensor networks,” in *15th International Parallel and Distributed Processing Symposium*, 2001, pp. 1965–1970.
- [99] M. Sichitiu and C. Veerarittiphan, “Simple, accurate time synchronization for wireless sensor networks,” in *IEEE Wireless Communications and Networking*, vol. 2, 2003, pp. 1266–1273.
- [100] A. Oppenheim, R. Schafer, and J. Buck, *Discrete-time signal processing*. Prentice Hall, Englewood Cliffs, NJ, 1998.

- [101] H. Liu and G. Xu, "Closed-form blind symbol estimation in digital communications," *IEEE Transactions on Signal Processing*, vol. 43, no. 11, pp. 2714–2723, 1995.
- [102] G. Xu, H. Liu, L. Tong, and T. Kailath, "A least-squares approach to blind channel identification," *IEEE Transactions on Signal Processing*, vol. 43, no. 12, pp. 2982–2993, 1995.
- [103] E. J. Cands and Y. Plan, "Near-ideal model selection by l1 minimization," *Annals of Statistics*, vol. 37, no. 5A, pp. 2145–2177, 2009.
- [104] H. Taylor, S. Banks, and J. McCoy, "Deconvolution with the l 1 norm," *Geophysics*, vol. 44, no. 1, pp. 39–52, 1979.
- [105] M. S. O'Brien, A. N. Sinclair, and S. M. Kramer, "Recovery of a sparse spike time series by L1 norm deconvolution," *IEEE Transactions on Signal Processing*, vol. 42, pp. 3353–3365, dec 1994.
- [106] O. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Survey of automatic modulation classification techniques: classical approaches and new trends," *IET Communications*, vol. 1, no. 2, pp. 137–156, April 2007.