Single Sensor Radio Scene Analysis for Packet Based Radio Signals

Goran Ivković, Predrag Spasojević, and Ivan Šeškar
WINLAB, ECE Department, Rutgers University
E-mail:{ ivkovic, spasojev, seskar}@winlab.rutgers.edu

Abstract— **We consider the problem of RF signal analysis where one sensing node observes a frequency band possibly used by multiple packet based radio transmitters. Analysis of the received signal consists of two steps. In the first step we use spectrogram to perform temporal segmentation of the received piecewise statistically homogeneous signal. This task is formulated as a clustering problem. In the second step we compute a certain 2-D slice of the fourth order spectrum for each of the segments found in the first step. The fourth order spectrum slices are arranged in a three-way array. Key idea of the second step is to use uniqueness properties of the low rank decomposition of the three-way array to recover spectra and associated activity sequences of individual components in the received signal. We derive a numerical algorithm for the low rank decomposition, which computes estimates of the spectra and activity sequences by optimizing a weighted least squares criterion under application specific constraints. The approach is illustrated with simulation examples involving signals used in 802.11a/b/g and Bluetooth networks. The proposed algorithm can be used as a spectrum analysis tool, providing crucial information needed for achieving efficient utilization of radio spectrum and elimination of mutual interference between the coexisting systems.**

I. Introduction

Most existing wireless systems use static allocation of radio spectrum. Each system operates in a predefined fixed frequency band. Since the allocated spectrum is not used at all locations and at all time instants, this static allocation leads to creation of spectrum holes, which implies inefficient utilization of the available radio spectrum. A much better spectrum utilization can be achieved with systems using dynamic spectrum allocation. These systems must have a completely new feature: spectrum sensing capability [1]. Spectrum sensing can be defined as RF signal analysis whose goal is to determine if the observed frequency band is occupied and also identify or characterize the signals present. Most of the existing work in spectrum sensing is motivated by its application in the emerging 802.22 standard, where the goal is to detect presence of so-called primary user, which in this case is digital TV signal in the 400-800 MHz frequency band [2]. This task is usually formulated as a binary hypothesis testing problem. Three main signal processing tools proposed for solving this problem and spectrum sensing in general are energy and related PSD based detectors, matched filter and cyclostationary detectors (see [1] [3] [4] and references in [4]).

This work was supported in part by a grant from TOYOTA InfoTechnology Center, U.S.A.

In this work we consider the spectrum sensing problems arising when the transmitters of interest form packet based radio networks. These networks are common in unlicensed frequency bands such as 2.4 GHz ISM and 5 GHz UNII bands, which are used by systems such as WLAN(802.11a/b/g), Bluetooth, Zig-Bee, cordless phones, etc. There are two application scenarios for spectrum sensing in this environment. In the first scenario, there is a communication network which identifies and exploits unused portions of the spectrum in the observed frequency band [5]. In the second scenario there is a dedicated spectrum sensing network. This concept is adopted in the emerging IEEE 1900.6 standard [4], which considers three types of systems: legacy systems using static spectrum allocation, a dedicated spectrum sensing network, and new systems using dynamic spectrum allocation. The role of the spectrum sensing network is to provide information about spectrum usage in the observed frequency band. This information can be used by the communication systems capable of dynamic spectrum allocation to access the available spectrum without causing interference to the legacy systems. Hence, the information provided by the spectrum sensing network could be crucial for achieving efficient utilization of radio spectrum. Spectrum sensing network could also be used for continuous monitoring of spectrum usage, detection and localization of transmitters violating established protocols for spectrum access, etc.

In the environment with packet based radio networks there are two key differences compared to the problems considered in the previous work. The first is the presence of multiple, possibly heterogeneous, transmitters, which in some cases transmit signals that overlap in time and frequency. The second difference is due to the fact that transmitters in packet based networks are regulated by their protocols and can transmit only in certain time slots. Thus, the transmitted signals are characterized by non-persistent excitation.

Let us now discuss applicability of existing approaches for spectrum sensing in this scenario. An obvious brute force approach would consist in building full receiver for each signal type. This approach would require complete knowledge of each transmitted signal format including the set of possible center frequencies. It is not only expensive and complicated, but also excessive since the goal here is to detect and identify the signals present and not to demodulate their transmitted information. A much simpler approach would be to use

preambles, which are sequences of known symbols embedded in packets to enable packet detection and synchronization. If these known sequences exist we can build matched filter detectors for them and thus, detect and identify signals present. Obviously we must build a separate detector for each signal format. This approach would also require the knowledge of possible center frequencies for each system. Its performance in an ideal case of a fully known signal in Gaussian noise is limited by the total signal energy [6], which in this application is limited by duration of the preamble. Finally, we note that presence of signals with non-persistent excitation prevents straightforward application of energy, PSD based, and cyclostationary detectors, since they all assume signals with persistent excitation during the observation interval.

We conclude that existing approaches do not offer a satisfactory solution in this scenario. Here, we present an entirely different approach. Each transmitted signal is characterized using its 2nd and 4th order statistics and an on/off activity sequence modeling its non-persistent excitation. The sensing node receives a superposition of these signals. As will be shown, using the 2nd and 4th order statistics of the received signal and certain uniqueness properties of low rank decomposition of three-way arrays, it is possible to identify spectra and activity sequences of individual signals forming the received signal. In other words, in contrast to existing spectrum sensing methods based on hypothesis testing(usually binary), we localize each signal in time and frequency by blind identification of its relevant parameters. The method is blind in a sense that the knowledge of signal pulse shapes and their center frequencies is not assumed.

II. Signal analysis method

We consider a setup where one sensing node observes a frequency band with M packet based radio sources. Now we make a number of assumptions. During the observation interval each source operates with constant power when on. However, the excitation in not persistent, which is modeled by an on/off activity sequence. We consider two types of transmitted signals: (1) linear modulation formats and (2) nonlinear modulation formats that can be approximated as a sum of finite number of signals in linear modulation formats. The transmitted signals can be characterized using their various statistical indicators. In this work we will use second order statistic(PSD) and partial information form fourth order statistic. Hence, each transmitted signal will be characterized by its second and fourth order spectra and an associated on/off activity sequence modelling its nonstationary behaviour. Received signal at the sensing node is a sum of transmitted signals and sensor noise. We assume that each transmitted signal is non-Gaussian and the sensor noise is stationary Gaussian signal with arbitrary PSD. Our approach performs analysis of the received signal in two steps. In the first step we perform temporal segmentation of the received nonstationary signal using its spectrogram. In the second step we use partial information from the fourth order spectra computed for the segments found in the first step to find spectra and activity sequences of individual signals forming the received signal.

A. Temporal segmentation algorithm

Under the stated assumptions the received signal consists of on/off contributions from various transmitted signals and stationary sensor noise. Thus, this signal is nonstationary consisting of a number of statistically homogeneous segments. Our goal is to determine those segments. We are going to assume that each segment is uniquely identified by its PSD and use spectrogram of the received signal for this task. Let T be the time resolution of the spectrogram. Assuming mutually uncorrelated source signals, the received PSD as a function of frequency f over the time interval $[(k-1)T, kT]$ is

$$S(f,k) = \sum_{m=1}^{M} |H_m(f)|^2 S_m(f) b_{km} + S_n(f) \qquad (1)$$

where $H_m(f)$ is the channel transfer function between the m-th source and the sensor, $S_m(f)$ is PSD of the m-th source and $b_{km} \in \{0,1\}$ indicates whether the m-th source is off or on during the time interval $[(k-1)T, kT]$. It is also assumed that the sensor noise is uncorrelated with source signals and has a PSD $S_n(f)$. The choice of T is the result of the inevitable trade-off between accuracy in tracking non-stationary changes of the source signals(a smaller T is better) and the achievable resolution in PSD estimation(a larger T is better). Let K measurements be performed over the observation interval of length KT. Let the total observed bandwidth be $W = J\Delta f$ where Δf is the frequency discretization step. We define the matrix $\boldsymbol{X}$ with entries

$$\boldsymbol{X}(j,k) = \int_{(j-1)\Delta f}^{j\Delta f} S(f,k)df = \sum_{m=1}^{M} a_{jm} b_{km} + n_j \qquad (2)$$

where $a_{jm} = \int_{(j-1)\Delta f}^{j\Delta f} |H_m(f)|^2 S_m(f)df$ for $j = 1, \ldots, J$, $J = W/\Delta f$ and, $n_j = \int_{(j-1)\Delta f}^{j\Delta f} S_n(f)df$. In an obvious matrix notation we have

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{n} \end{bmatrix} \begin{bmatrix} \boldsymbol{B} & \boldsymbol{1} \end{bmatrix}^T \qquad (3)$$

where $\boldsymbol{A} = [a_{jm}] = \begin{bmatrix} \boldsymbol{a}_1 & \ldots & \boldsymbol{a}_M \end{bmatrix}$, $\boldsymbol{n} = [n_j]$, and $\boldsymbol{B} = [b_{km}]$. Let $\hat{\boldsymbol{X}}$ be an estimate of $\boldsymbol{X}$, where each column $\hat{\boldsymbol{x}}$ is computed as periodogram with frequency smoothing [7]. We observe that the columns of $\hat{\boldsymbol{X}}$ form clusters, where each cluster(segment) is defined as a union of time intervals with the same combination of active sources. For example, the cluster centroids could be $\boldsymbol{n}$, $\boldsymbol{a}_1 + \boldsymbol{n}$, $\boldsymbol{a}_1 + \boldsymbol{a}_2 + \boldsymbol{n}$, etc. Hence, the segmentation problem is in fact a clustering problem. We model the columns of $\hat{\boldsymbol{X}}$ using the mixture pdf model

$$f(\hat{\boldsymbol{x}}) = \sum_{i=1}^{N} p_i g(\hat{\boldsymbol{x}} | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \qquad (4)$$

where N is the number of clusters, p_i is the probability of the i-th cluster, and $g(\hat{\boldsymbol{x}}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ is a multivariate Gaussian pdf with mean $\boldsymbol{\mu}_i$ and covariance $\boldsymbol{\Sigma}_i$. Dispersion around cluster means is caused by inevitable PSD estimation errors due to

the finite sample size effects. True distribution of these errors is signal dependent and hence, unknown in our application. However, we estimate each PSD value as an average value over several neighbouring frequencies and hence, it makes sense to assume that the estimation errors are approximately Gaussian. Assuming the model given by (4) the clustering problem reduces to the estimation of the parameters p_i, $\boldsymbol{\mu}_i$, and $\boldsymbol{\Sigma}_i$ for $i = 1, \ldots, N$. Maximum likelihood estimates of these parameters can be obtained using EM algorithm as follows [8]. Let us define the cluster membership matrix $\boldsymbol{W} = [w_{ik}]$, where $w_{ik} = 1$ if the k-th column of $\hat{\boldsymbol{X}}$ belongs to the i-th cluster, otherwise $w_{ik} = 0$. Assuming the model parameters are known w_{ik} can be estimated as its expected value, which can be found using the Bayes' rule as

$$\hat{w}_{ik} = P(C_i|\hat{\boldsymbol{x}}_k) = \frac{p_i g(\hat{\boldsymbol{x}}_k|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\sum_{i=1}^{N} p_i g(\hat{\boldsymbol{x}}_k|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}. \tag{5}$$

where $P(C_i|\hat{\boldsymbol{x}}_k)$ is the probability that the k-th column of $\hat{\boldsymbol{X}}$ belongs to the i-th cluster. Assuming now that $\boldsymbol{W}$ is known the ML estimates of the model parameters are given as [8]

$$\hat{p}_i = \frac{1}{K}\sum_{k=1}^{K} w_{ik}, \;\; \hat{\boldsymbol{\mu}}_i = \frac{\sum_{k=1}^{K} w_{ik}\hat{\boldsymbol{x}}_k}{\sum_{k=1}^{K} w_{ik}}, \tag{6}$$

and

$$\hat{\boldsymbol{\Sigma}}_i = \frac{\sum_{k=1}^{K} w_{ik}(\hat{\boldsymbol{x}}_k - \hat{\boldsymbol{\mu}}_i)(\hat{\boldsymbol{x}}_k - \hat{\boldsymbol{\mu}}_i)^T}{\sum_{k=1}^{K} w_{ik}}. \tag{7}$$

We start from random initial estimates for the model parameters and then use equations (5)-(7) until convergence. The algorithm converges monotonically to a local maximum of the likelihood function of the model (4). The global maximum can be found by repeatedly using the algorithm from different initial points. This concludes the segmentation procedure. Now we would like to know what combination of transmitted signals was present in addition to sensor noise in each of the identified signal segments. It is difficult to answer this question reliably using only PSD information. As it will become clear, we can get a satisfactory answer to this question by using information contained in fourth order spectrum.

B. Signal analysis using fourth order spectrum

Let us first introduce a few definitions. Let $x(t)$ be a random process admitting Cramer spectral representation $x(t) = \int_{-\infty}^{\infty} e^{j2\pi f} dX(f)$ where $dX(f)$ is called spectral process [9]. The second order spectrum(PSD) of $x(t)$ is defined as $S_{2x}(f)df = E[|dX(f)|^2]$ and the fourth order spectrum(trispectrum) is defined as [9]

$$\begin{aligned}
&S_{4x}(f_1, f_2, f_3)df_1df_2df_3 = \\
&cum(dX(f_1), dX(f_2), dX(-f_3)^*, dX(f_1+f_2+f_3)^*) = \\
&E[dX(f_1)dX(f_2)dX(-f_3)^*dX(f_1+f_2+f_3)^*] \\
&- E[dX(f_1)dX(f_2)]E[dX(-f_3)^*dX(f_1+f_2+f_3)^*] \\
&- E[dX(f_1)dX(-f_3)^*]E[dX(f_2)dX(f_1+f_2+f_3)^*] \\
&- E[dX(f_1)dX(f_1+f_2+f_3)^*]E[dX(f_2)dX(-f_3)^*]
\end{aligned} \tag{8}$$

where we used the definition of the fourth order cumulants [10]. In our work we are going to use the following 2-D slice of the trispectrum $\tilde{S}_{4x}(f, v) = S_{4x}(f, v, -v)$ which using the definition (8) can be written as

$$\tilde{S}_{4x}(f, v)dfdv = cum(dX(f), dX(v), dX(v)^*, dX(f)^*). \tag{9}$$

The quantity $\tilde{S}_{4x}(f, v)$ for $f = v$ is the 4-th 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. The quantity $\tilde{S}_{4x}(f, v)$ for $f \neq v$ measures a certain form of statistical dependency between spectral components at frequencies f and v. In contrast, the PSD $S_{2x}(f)$ measures only average power at frequency f and contains no information about statistical dependencies between spectral components at different frequencies. Hence, the slice $\tilde{S}_{4x}(f, v)$ contains significant information about the received signal, which is not present in its PSD.

Let $\tilde{S}_4(f, v, i)$ be the trispectrum slice defined above for the i-th segment(cluster) found in the first step of the algorithm. Assuming statistically independent signals we have

$$\tilde{S}_4(f, v, i) = \sum_{m=1}^{M} |H_m(f)|^2 |H_m(v)|^2 \tilde{S}_{4m}(f, v)c_{im} + \tilde{S}_{4n}(f, v) \tag{10}$$

where $\tilde{S}_{4m}(f, v)$ is the trispectrum slice of the m-th source and $c_{im} \in \{0, 1\}$ indicates whether the m-th source is on in the i-th segment for $i = 1, \ldots, N$. We define $\boldsymbol{C} = [c_{im}]$ and note that $\boldsymbol{B} = \boldsymbol{W}^T\boldsymbol{C}$, which follows from the definition of $\boldsymbol{W}$ given in Section II.A. Since we assumed the receiver noise is Gaussian, we have $\tilde{S}_{4n}(f, v) = 0$. Hence, using the fourth order spectrum we can discriminate between communication signals which are practically always non-Gaussian and Gaussian receiver noise. Let us define a three-way array $\boldsymbol{Y}$ with entries

$$\boldsymbol{Y}(j, n, i) = \int_{(j-1)\Delta f}^{j\Delta f}\int_{(n-1)\Delta f}^{n\Delta f} \tilde{S}_4(f, v, i)dfdv = \sum_{m=1}^{M} q_{jnm}c_{im} \tag{11}$$

where $q_{jnm} = \int_{(j-1)\Delta f}^{j\Delta f}\int_{(n-1)\Delta f}^{n\Delta f} |H_m(f)|^2|H_m(v)|^2$ $\tilde{S}_{4m}(f, v)dfdv$ for $j, n = 1, \ldots, J$. As defined earlier $J = W/\Delta f$, where W is the total observed bandwidth. Let $\boldsymbol{Y}_i$ be the J by J matrix obtained by fixing the index i in (11). We have

$$\boldsymbol{Y}_i = \sum_{m=1}^{M} \boldsymbol{Q}_m c_{im} \tag{12}$$

where $\boldsymbol{Q}_m = [q_{jnm}]$ is the matrix characterizing the m-th source. 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 source signal has the following form

$$x_m(t) = \sum_{k=-\infty}^{\infty} a_{km}p_m(t - kT_{sm}) \tag{13}$$

where a_{km} is an iid sequence of input symbols, $p_m(t)$ is the pulse shape, and T_{sm} is the symbol period. Using the definition it can be shown that the trispectrum of $x_m(t)$ is

$$S_{4m}(f_1,f_2,f_3) = \frac{S_{4am}(f_1,f_2,f_3)}{T_{sm}} P_m(f_1)P_m(f_2) P_m(-f_3)^* P_m(f_1+f_2+f_3)^* \quad (14)$$

where $P_m(f) = \int_{-\infty}^{\infty} p_m(t)e^{-j2\pi ft}dt$. Since a_{km} is an iid sequence $S_{4am}(f_1,f_2,f_3) = \gamma_{4am} = cum(a_{km},a_{km},a_{km}^*,a_{km}^*)$. Using (9) we have

$$\tilde{S}_{4m}(f,v) = k_m|P_m(f)|^2|P_m(v)|^2 \quad (15)$$

where $k_m = \gamma_{4am}/T_{sm}$. 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 $\boldsymbol{Q}_m$ is of rank one. In this case the entries of $\boldsymbol{Y}$ are

$$\boldsymbol{Y}(j,n,i) = \sum_{m=1}^{M} k_m f_{jm} f_{nm} c_{im}. \quad (16)$$

Now we observe that

$$\boldsymbol{Y}_i = \boldsymbol{F}\boldsymbol{\Lambda}_i\boldsymbol{F}^T \quad (17)$$

where $\boldsymbol{F} = [f_{jm}]$ and $\boldsymbol{\Lambda}_i = diag(\begin{bmatrix} k_1c_{i1} & \dots & k_Mc_{iM}\end{bmatrix})$. We see that the matrix $\boldsymbol{F}$ jointly diagonalizes the matrices $\boldsymbol{Y}_i$ for $i = 1,\dots,N$. The problem of finding $\boldsymbol{F}$ from the set of matrices $\boldsymbol{Y}_i$ for $i = 1,\dots,N$ is known as joint diagonalization by congruence. It is well known that, under some mild conditions, the matrices $\boldsymbol{F}$ and $\boldsymbol{C} = [c_{im}]$ can be determined up to a scaling and permutation of their columns by joint diagonalization of the matrices $\boldsymbol{Y}_i$ for $i = 1,\dots,N$ [11]. This is the key idea of our method. Several important blind signal separation methods also reduce to joint diagonalization problems(see references in [12] and [13]). Let us explain this key idea from another point of view. We observe that (16) represents decomposition of $\boldsymbol{Y}$ into M three-way rank-one terms [14]. This decomposition is unique under certain algebraic conditions [15]. When the uniqueness conditions hold the rank-one terms in (16) are *uniquely* determined. In other words, we can uniquely determine terms in the sum in (16), which represent contributions of different transmitted signals to the observed three-way array $\boldsymbol{Y}$. Thus, the decomposition of $\boldsymbol{Y}$ into rank-one terms becomes a parameter estimation tool.

Nonlinear modulation formats. When the source signals are in nonlinear modulation formats(e. g., FM signals), we cannot put the trispectrum slices $\tilde{S}_{4m}(f,v)$ in the form given by (15). In that case $rank(\boldsymbol{Q}_m) > 1$ and the above procedure cannot be applied. Obviously, nonlinearity is not a property and therefore, it cannot be treated without introducing additional assumptions. Let us assume that the m-th source signal can be represented or approximated as a sum of R_m signals in linear format given by (13). Then its trispectrum slice can be represented as

$$\boldsymbol{Q}_m = \sum_{r=1}^{R_m} k_{rm}\boldsymbol{f}_r^{(m)}\boldsymbol{f}_r^{(m)T} = \boldsymbol{F}_m\boldsymbol{D}_m\boldsymbol{F}_m^T \quad (18)$$

where $\boldsymbol{D}_m = diag(\begin{bmatrix} k_{1m} & \dots & k_{R_m,m}\end{bmatrix})$ contains constants defined in (15) and $\boldsymbol{F}_m = \begin{bmatrix}\boldsymbol{f}_1^{(m)} & \dots & \boldsymbol{f}_{R_m}^{(m)}\end{bmatrix}$ contains the associated squared magnitude frequency responses of the linear signals used to represent the m-th source signal. The entries of $\boldsymbol{Y}$ can be written as

$$\boldsymbol{Y}(j,n,i) = \sum_{m=1}^{M} q_{jnm}c_{im} = \sum_{m=1}^{M}\sum_{r=1}^{R_m} k_{rm}f_{jrm}f_{nrm}c_{im} \quad (19)$$

where $\boldsymbol{F}_m(j,r) = f_{jrm}$. Equation (19) represents decomposition of $\boldsymbol{Y}$ into M block terms. Now each signal contributes one block term rather than one rank-one term, which was the case with linear modulation formats. Using the operator $vecr$ which transforms real symmetric matrix into a vector [13], the decomposition into block terms can be represented as

$$\boldsymbol{Y}_V = \begin{bmatrix} vecr(\boldsymbol{Y}_1) & \dots & vecr(\boldsymbol{Y}_N)\end{bmatrix} = \boldsymbol{F}_V\boldsymbol{K}\boldsymbol{C}^T \quad (20)$$

where

$$\boldsymbol{F}_V = [vecr(\boldsymbol{f}_1^{(1)}\boldsymbol{f}_1^{(1)T}) \quad \dots \quad vecr(\boldsymbol{f}_{R_1}^{(1)}\boldsymbol{f}_{R_1}^{(1)T}) \quad \dots \quad vecr(\boldsymbol{f}_1^{(M)}\boldsymbol{f}_1^{(M)T}) \quad \dots \quad vecr(\boldsymbol{f}_{R_M}^{(M)}\boldsymbol{f}_{R_M}^{(M)T})], \quad (21)$$

$$\boldsymbol{K} = \begin{bmatrix} k_{11} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ k_{R_1,1} & 0 & \dots & 0 \\ 0 & k_{12} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & k_{R_2,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & k_{1,M} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & k_{R_M,M} \end{bmatrix}, \quad (22)$$

and $\boldsymbol{C} = [c_{im}]$ is an N by M matrix defined earlier with entries $c_{im} \in \{0,1\}$. We note that decomposition into rank-one terms is a special case of the decomposition into block terms obtained for $R_1 = \dots = R_M = 1$. The decomposition into block terms can also be unique under certain algebraic conditions [16], which are generalization of those from [15]. When these conditions hold block terms in (19), representing contributions of different transmitted signals to $\boldsymbol{Y}$, are *uniquely* determined and hence, the decomposition of $\boldsymbol{Y}$ into block terms becomes a parameter estimation tool.

It should be mentioned that uniqueness conditions derived in [15] and [16] do not take into account our application specific constraints: symmetry($\boldsymbol{Y}_i = \boldsymbol{Y}_i^T$), nonnegativity($f_{jrm} \geq 0$), and finite alphabet($c_{im} \in \{0,1\}$). Hence, uniqueness conditions in our application are probably even less restrictive than those from [15] and [16]. However, it is very difficult to find minimal uniqueness conditions in this application and express them in some easy to check form. As will be demonstrated with concrete examples in section IV, decomposition of $\boldsymbol{Y}$ into rank-one or block terms is indeed a useful signal analysis tool

in some important practical situations. Another important point is that the uniqueness conditions from [15] and [16] do not hold for two-way arrays(i. e., matrices). This fact explains why we could not solve the problem using second order spectra and had to use fourth order spectra.

III. Computational methods

In this section we develop algorithms needed to compute an estimate of the three-way $\boldsymbol{Y}$ and to decompose it into block terms, which represent contributions of individual signals present in the received signal.

A. Estimation of fourth order spectra

First, we need to compute estimates of the trispectrum matrices $\boldsymbol{Y}_i$ for each of the N temporal segments found in the first step. Let $\boldsymbol{y}_k$ be the column vector computed by applying FFT on the received signal samples over the k-th time interval defined in (1) for $k = 1, \ldots, K$. We compute an estimate of $\boldsymbol{Y}_i$ by replacing $dX(f)$ in (9) with the corresponding entries from $\boldsymbol{y}_k$ and performing averaging over the i-th temporal segment

$$\begin{aligned}\hat{\boldsymbol{Y}}_i = &\frac{1}{\tau_i}\sum_{k=1}^{K} \hat{w}_{ik}(\boldsymbol{y}_k \bullet \boldsymbol{y}_k^*)(\boldsymbol{y}_k \bullet \boldsymbol{y}_k^*)^T \\ &- \left(\frac{1}{\tau_i}\sum_{k=1}^{K} \hat{w}_{ik}\boldsymbol{y}_k\boldsymbol{y}_k^T\right) \bullet \left(\frac{1}{\tau_i}\sum_{k=1}^{K} \hat{w}_{ik}\boldsymbol{y}_k\boldsymbol{y}_k^T\right)^* \\ &- \left(\frac{1}{\tau_i}\sum_{k=1}^{K} \hat{w}_{ik}\boldsymbol{y}_k\boldsymbol{y}_k^H\right) \bullet \left(\frac{1}{\tau_i}\sum_{k=1}^{K} \hat{w}_{ik}\boldsymbol{y}_k\boldsymbol{y}_k^H\right)^* \\ &- \left(\frac{1}{\tau_i}\sum_{k=1}^{K} \hat{w}_{ik}(\boldsymbol{y}_k \bullet \boldsymbol{y}_k^*)\right)\left(\frac{1}{\tau_i}\sum_{k=1}^{K} \hat{w}_{ik}(\boldsymbol{y}_k \bullet \boldsymbol{y}_k^*)^T\right)\end{aligned} \tag{23}$$

where $\bullet$ denotes elementwise(Hadamard) matrix product, $\tau_i = \sum_{k=1}^{K} \hat{w}_{ik}$, and $\hat{w}_{ik}$ are estimates from (5) computed using (5)-(7). The estimate $\hat{\boldsymbol{Y}}_i$ is computed by temporal averaging. Additional frequency averaging can be done by applying a 2D interpolation filter on $\hat{\boldsymbol{Y}}_i$. The estimates of fourth order spectra computed over the determined temporal segments have much smaller variance compared to the corresponding estimates computed over the time intervals of duration T defined in (1). This fact explains why temporal segmentation was performed as the first step.

B. Determining the number of rank-one and block terms

Our final objective is to develop an algorithm for decomposition of $\boldsymbol{Y}$ into block terms given by (19) and (20). In order to do that we need to know the number of block terms M and the number of rank-one terms in each of the block terms $R_1, \ldots, R_M$. In this subsection we describe how (at least in theory) the parameters M and $R = R_1 + \cdots + R_M$ can be determined from $\boldsymbol{Y}$. First, we apply $vecr$ operator on (12) for $i = 1, \ldots, N$. We have

$$\begin{bmatrix} vecr(\boldsymbol{Y}_1) & \ldots & vecr(\boldsymbol{Y}_N) \end{bmatrix} = \boldsymbol{Q}_V \boldsymbol{C}^T \tag{24}$$

where

$$\boldsymbol{Q}_V = \begin{bmatrix} vecr(\boldsymbol{Q}_1) & \ldots & vecr(\boldsymbol{Q}_M) \end{bmatrix}. \tag{25}$$

Assuming that $J + \binom{J}{2} \geq M$, $N \geq M$, and that $\boldsymbol{Q}_V$ and $\boldsymbol{C}$ are of full rank we have

$$rank\left(\begin{bmatrix} vecr(\boldsymbol{Y}_1) & \ldots & vecr(\boldsymbol{Y}_N) \end{bmatrix}\right) = M \tag{26}$$

which can be used to determine M. Next, we show how $R = R_1 + \cdots + R_M$ can be found from $\boldsymbol{Y}$. Using (12) and (18) we have

$$\boldsymbol{Y}_i = \boldsymbol{F}_A \boldsymbol{L}_i \boldsymbol{F}_A^T \tag{27}$$

where

$$\boldsymbol{F}_A = \begin{bmatrix} \boldsymbol{F}_1 & \ldots & \boldsymbol{F}_M \end{bmatrix} \tag{28}$$

and

$$\boldsymbol{L}_i = diag(\boldsymbol{L}(i,:)) \tag{29}$$

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

$$\boldsymbol{L} = \begin{bmatrix} k_{11}c_{11} & \ldots & k_{11}c_{N1} \\ \vdots & \ddots & \vdots \\ k_{R_1,1}c_{11} & \ldots & k_{R_1,1}c_{N1} \\ \vdots & \ddots & \vdots \\ k_{1M}c_{1M} & \ldots & k_{11}c_{NM} \\ \vdots & \ddots & \vdots \\ k_{R_M,M}c_{1M} & \ldots & k_{R_M,M}c_{NM} \end{bmatrix}^T. \tag{30}$$

It is very simple to check that

$$\begin{bmatrix} \boldsymbol{Y}_1 & \ldots & \boldsymbol{Y}_N \end{bmatrix} = \boldsymbol{F}_A(\boldsymbol{L} \odot \boldsymbol{F}_A)^T \tag{31}$$

where the symbol $\odot$ denotes Khatri-Rao product defined as [14]

$$\boldsymbol{L} \odot \boldsymbol{F}_A = \begin{bmatrix} \boldsymbol{L}(:,1) \otimes \boldsymbol{F}_A(:,1) & \ldots & \boldsymbol{L}(:,R) \otimes \boldsymbol{F}_A(:,R) \end{bmatrix} \tag{32}$$

where the symbol $\otimes$ denotes Kronecker product. Assuming that $J \geq R$, $rank(\boldsymbol{F}_A) = R$, and $rank(\boldsymbol{L} \odot \boldsymbol{F}_A) = R$ we have

$$rank\left(\begin{bmatrix} \boldsymbol{Y}_1 & \ldots & \boldsymbol{Y}_N \end{bmatrix}\right) = R \tag{33}$$

which can be used to determine R.

If $\boldsymbol{Y}$ is perfectly known M and R can be found from SVDs of the matrices in (26) and (33). In any practical situation only an estimate $\hat{\boldsymbol{Y}}$, containing some amount of error, is available. In that case M and R can be found as effective ranks of the matrices given by (26) and (33). These problems are very difficult themselves and require separate algorithms(e. g., [17]), which are beyond the scope of this paper. In the simulation examples presented in Section IV the parameters M and R are assumed to be known.

C. Decomposition into rank-one terms

In this subsection we develop an algorithm for decomposition of $\boldsymbol{Y}$ into rank-one terms given by (16). We have seen that this decomposition is equivalent to joint diagonalization of the set of matrices $\boldsymbol{Y}_i$ for $i = 1, \ldots, N$. There are several numerical algorithms for joint diagonalization problems [13] [18] [12], but they do not take into account our application specific constraints. In [19] and [20] two methods for decomposition of three-way arrays into rank-one terms under optional nonnegativity constraints are presented. However, these methods are not concise algorithms that can be implemented and tried, but very complicated optimization programs where the author made numerous implementation decisions. Hence, these methods do not provide a satisfactory solution for our problem. Here, we develop an algorithm specifically for this application. In order to compute the decomposition given by (16), we form the following weighted least squares criterion

$$C_{wls} = \sum_{j=1}^{J} \sum_{n=j}^{J} \sum_{i=1}^{N} \omega_i \left(\hat{\boldsymbol{Y}}(j,n,i) - \sum_{m=1}^{M} k_m f_{jm} f_{nm} c_{im} \right)^2. \tag{34}$$

where $\hat{\boldsymbol{Y}}(j,n,i)$ is the observed(estimated) value given by $\hat{\boldsymbol{Y}}(j,n,i) = \boldsymbol{Y}(j,n,i) + \boldsymbol{E}(j,n,i)$ where $\boldsymbol{E}(j,n,i)$ is the estimation error. We also used the symmetry $\hat{\boldsymbol{Y}}(j,n,i) = \hat{\boldsymbol{Y}}(n,j,i)$. One reasonable choice for weights is $\omega_i = \hat{p}_i$, since each matrix $\hat{\boldsymbol{Y}}_i$ is estimated from a cluster with probability p_i as defined in (4). We seek parameters $\boldsymbol{F} = [f_{jm}]$, $\boldsymbol{C} = [c_{im}]$, and $\boldsymbol{k} = \begin{bmatrix} k_1 & \ldots & k_M \end{bmatrix}^T$ that minimize C_{wls} subject to the constraints $f_{jm} \geq 0$ and $c_{im} \in \{0, 1\}$. First, we will solve the problem where the second constraint is relaxed to $c_{im} \geq 0$. We will explain in the next section how to impose $c_{im} \in \{0, 1\}$. We seek parameter estimates satisfying

$$[\hat{\boldsymbol{F}}, \hat{\boldsymbol{C}}, \hat{\boldsymbol{k}}] = \underset{\boldsymbol{F} \geq 0, \boldsymbol{C} \geq 0, \boldsymbol{k}}{\operatorname{argmin}} C_{wls}(\boldsymbol{F}, \boldsymbol{C}, \boldsymbol{k}). \tag{35}$$

Next, we apply certain tools for numerical optimization, discussed in [21] and [22], on problem (35). Our first step is to take into account nonnegativity constraints using logarithmic barrier function, which is a well known method [21, ch. 11]. This method was also used in [19]. We define new criterion

$$C_{bf} = C_{wls} - \alpha \left(\sum_{j=1}^{J} \sum_{m=1}^{M} \log f_{jm} + \sum_{i=1}^{N} \sum_{m=1}^{M} \log c_{im} \right) \tag{36}$$

where α is a small positive constant, which regulates influence of the added logarithmic terms. When any of the parameters with nonnegative constraint, say f_{jm}, 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. Now we can consider the unconstrained problem

$$[\hat{\boldsymbol{F}}, \hat{\boldsymbol{C}}, \hat{\boldsymbol{k}}] = \underset{\boldsymbol{F}, \boldsymbol{C}, \boldsymbol{k}}{\operatorname{argmin}} C_{bf}(\boldsymbol{F}, \boldsymbol{C}, \boldsymbol{k}). \tag{37}$$

Let us try to solve (37) using Newton's method. We define the parameter vector

$$\boldsymbol{p} = \begin{bmatrix} vec(\boldsymbol{F})^T & vec(\boldsymbol{C})^T & \boldsymbol{k}^T \end{bmatrix}^T. \tag{38}$$

Starting from an initial point we update parameter vector $\boldsymbol{p}$ in each step as follows. Let $\boldsymbol{p}_0$ be the current parameter vector. We approximate the objective function around the current point using quadratic approximation

$$C_{bf}(\boldsymbol{p}_0 + \Delta\boldsymbol{p}) \approx C_{bf}(\boldsymbol{p}_0) + \boldsymbol{g}^T \Delta\boldsymbol{p} + \frac{1}{2} \Delta\boldsymbol{p}^T \boldsymbol{H} \Delta\boldsymbol{p} \tag{39}$$

where $\boldsymbol{g}$ is gradient and $\boldsymbol{H}$ is Hessian of $C_{bf}(\boldsymbol{p})$. We find the step $\Delta\boldsymbol{p}$ by solving

$$\min_{\Delta\boldsymbol{p}} \boldsymbol{g}^T \Delta\boldsymbol{p} + \frac{1}{2} \Delta\boldsymbol{p}^T \boldsymbol{H} \Delta\boldsymbol{p} \tag{40}$$

For positive definite $\boldsymbol{H}$ the optimal step is found from [21]

$$\boldsymbol{g} + \boldsymbol{H} \Delta\boldsymbol{p} = \boldsymbol{0} \tag{41}$$

and the updated parameter vector is

$$\boldsymbol{p} = \boldsymbol{p}_0 + \Delta\boldsymbol{p}. \tag{42}$$

However, there is a problem in direct application of Newton's method on minimization of $C_{bf}(\boldsymbol{p})$. As mentioned previously the columns of $\boldsymbol{F}$ and $\boldsymbol{C}$ are determined up to arbitrary scaling factors. This means that there is no unique vector $\boldsymbol{p}$ that minimizes $C_{bf}(\boldsymbol{p})$. In other words, the problem of minimizing $C_{bf}(\boldsymbol{p})$ has a continuum of solutions rather than one solution. In vicinity of any of these solutions there is no unique solution for $\Delta\boldsymbol{p}$ in (41) and hence, $\boldsymbol{H}$ must be singular. This problem is solved by imposing unit length constraints on the columns of $\boldsymbol{F}$ and $\boldsymbol{C}$. Since the entries are nonnegative we can impose the following constraints: $\sum_{j=1}^{J} f_{jm} = 1$ and $\sum_{i=1}^{N} c_{im} = 1$ for $m = 1, \ldots, M$. These constraints can be expressed as $\boldsymbol{A}_{eq}\boldsymbol{p} = \boldsymbol{1}$ where $\boldsymbol{A}_{eq}$ is a $2M$ by $(J + N + 1)M$ matrix of zeros and ones. Now we need to minimize $C_{bf}(\boldsymbol{p})$ subject to linear constraints $\boldsymbol{A}_{eq}\boldsymbol{p} = \boldsymbol{1}$, which can be done using methods from [21, ch. 10]. It follows from (42) $\boldsymbol{A}_{eq}\boldsymbol{p} = \boldsymbol{A}_{eq}(\boldsymbol{p}_0 + \Delta\boldsymbol{p})$. Since the equality constraint must be satisfied in every iteration, we also have $\boldsymbol{A}_{eq}\boldsymbol{p} = \boldsymbol{A}_{eq}\boldsymbol{p}_0 = \boldsymbol{1}$, which implies $\boldsymbol{A}_{eq}\Delta\boldsymbol{p} = \boldsymbol{0}$. Now we need to solve (40) subject to $\boldsymbol{A}_{eq}\Delta\boldsymbol{p} = \boldsymbol{0}$. The optimal step is found by solving [21, pp. 526]

$$\begin{bmatrix} \boldsymbol{H} & \boldsymbol{A}_{eq}^T \\ \boldsymbol{A}_{eq} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{p} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{g} \\ \boldsymbol{0} \end{bmatrix} \tag{43}$$

where $\boldsymbol{\lambda}$ is a vector of Lagrange multipliers. This procedure is equivalent to elimination of equality constraints and performing the minimization in (40) over $\Delta\boldsymbol{p} \in Null(\boldsymbol{A}_{eq})$. Finally, we need to compute gradient $\boldsymbol{g}$ and Hessian $\boldsymbol{H}$ of the criterion function $C_{bf}(\boldsymbol{p})$ given by (36). Let us define the vector

$$\hat{\boldsymbol{y}} = \begin{bmatrix} vecr(\hat{\boldsymbol{Y}}_1)^T & \ldots & vecr(\hat{\boldsymbol{Y}}_N)^T \end{bmatrix}^T \tag{44}$$

and in the same way $\boldsymbol{y}(\boldsymbol{p})$ as a result of vectorization of $\boldsymbol{Y}$, where we emphasised the functional dependence on the parameter vector. The functional dependence is given by (16) and (38). Now we have

$$C_{wls}(\boldsymbol{p}) = \sum_{q=1}^{Q} w_q \left(\hat{y}_q - y_q(\boldsymbol{p})\right)^2 \tag{45}$$

where $Q = N(J + \binom{J}{2})$ and $\hat{y}_q$ and $y_q(\boldsymbol{p})$ are q-th entries of $\hat{\boldsymbol{y}}$ and $\boldsymbol{y}(\boldsymbol{p})$ respectively, and weights w_q are diagonal entries of the matrix

$$\boldsymbol{\Gamma} = diag\left(\begin{bmatrix}\omega_1 & \dots & \omega_1 & \dots & \omega_N & \dots & \omega_N\end{bmatrix}\right). \tag{46}$$

Using (36) and (45) we find the gradient of $C_{bf}(\boldsymbol{p})$ as

$$\boldsymbol{g}(\boldsymbol{p}) = -2\boldsymbol{J}(\boldsymbol{p})^T\boldsymbol{\Gamma}(\hat{\boldsymbol{y}} - \boldsymbol{y}(\boldsymbol{p})) - \alpha\boldsymbol{g}_{log}(\boldsymbol{p}) \tag{47}$$

where $\boldsymbol{J}(\boldsymbol{p})$ is the Jacobian matrix [22] of $\boldsymbol{y}(\boldsymbol{p})$ with entries $j_{qr}(\boldsymbol{p}) = \frac{\partial y_q(\boldsymbol{p})}{\partial p_r}$ and

$$\boldsymbol{g}_{log}(\boldsymbol{p}) = \begin{bmatrix}f_{11}^{-1} & \dots & f_{JM}^{-1} & \dots & c_{11}^{-1} & \dots & c_{NM}^{-1} & 0 & \dots & 0\end{bmatrix}^T. \tag{48}$$

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

$$\boldsymbol{H}(\boldsymbol{p}) = 2\boldsymbol{J}(\boldsymbol{p})^T\boldsymbol{\Gamma}\boldsymbol{J}(\boldsymbol{p}) - \sum_{q=1}^{Q} w_q \left(\hat{y}_q - y_q(\boldsymbol{p})\right)\boldsymbol{G}_q(\boldsymbol{p}) + \alpha\boldsymbol{H}_{log}(\boldsymbol{p}) \tag{49}$$

where $\boldsymbol{G}_q(\boldsymbol{p})$ is Hessian of the function $y_q(\boldsymbol{p})$ and

$$\boldsymbol{H}_{log}(\boldsymbol{p}) = diag(\begin{bmatrix}f_{11}^{-2} & \dots & f_{JM}^{-2} & \dots & c_{11}^{-2} & \dots & c_{NM}^{-2} & 0 & \dots & 0\end{bmatrix}). \tag{50}$$

Newton's method with equality constraints can be used only if the Hessian $\boldsymbol{H}(\boldsymbol{p})$ given by (49) is positive definite on $Null(\boldsymbol{A}_{eq})$. We see that the second term on the right hand side of (49) is in general indefinite and hence, $\boldsymbol{H}(\boldsymbol{p})$ is also indefinite. Now we have two options. The first is to use modified Newton's methods, where indefinite Hessian $\boldsymbol{H}(\boldsymbol{p})$ is replaced with a positive definite matrix, which is close to the original Hessian in certain sense [22, sec. 4.4.2]. The second option is to simply ignore the second term in (49) and approximate the Hessian as the sum of the first and third term, which produces a positive definite matrix. This approach is sometimes called Gauss-Newton method [22, sec. 4.7.2]. We experimented with these two approaches and found that both of them lead to reliable algorithms. Finally, we decided to pick Gauss-Newton method since it requires fewer computations. The Hessian approximated as

$$\boldsymbol{H}(\boldsymbol{p}) \approx 2\boldsymbol{J}(\boldsymbol{p})^T\boldsymbol{\Gamma}\boldsymbol{J}(\boldsymbol{p}) + \alpha\boldsymbol{H}_{log}(\boldsymbol{p}) \tag{51}$$

is used to compute the step $\Delta\boldsymbol{p}$ using (43). With positive definite Hessian on $Null(\boldsymbol{A}_{eq})$ the computed step $\Delta\boldsymbol{p}$ represents a descent direction [22]. However, using (42) to compute the updated parameter vector may not lead to $C_{bf}(\boldsymbol{p}) \leq C_{bf}(\boldsymbol{p}_0)$, since the updated vector is based on the quadratic approximation (39). Also the logarithmic barrier function is approximated in the same way and hence, using (42) may violate nonnegativity constraints. These problems are resolved by introducing procedure for step length control [22], which is also known as line search [21]. Here, we use a very simple line search where we compute the updated parameter vector as

$$\boldsymbol{p} = \boldsymbol{p}_0 + \mu^k\Delta\boldsymbol{p}. \tag{52}$$

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

Let us summarize the algorithm for decomposition into rank-one terms(joint diagonalization). The algorithm computes parameter estimates

$$[\hat{\boldsymbol{F}}, \hat{\boldsymbol{C}}, \hat{\boldsymbol{k}}] = \underset{\boldsymbol{F}\geq 0, \boldsymbol{1}^T\boldsymbol{F}=\boldsymbol{1}^T, \boldsymbol{C}\geq 0, \boldsymbol{1}^T\boldsymbol{C}=\boldsymbol{1}^T, \boldsymbol{k}}{\operatorname{argmin}} C_{wls}(\boldsymbol{F}, \boldsymbol{C}, \boldsymbol{k}) \tag{53}$$

as follows. Starting from a random initial point we iteratively update the parameter vector $\boldsymbol{p}$ defined by (38). In each iteration we compute $\boldsymbol{J}(\boldsymbol{p})$ as partial derivatives of model equations (16), $\boldsymbol{H}(\boldsymbol{p})$ using (51), and $\boldsymbol{g}(\boldsymbol{p})$ using (47). Then we compute $\Delta\boldsymbol{p}$ by solving equations (43) and update the parameter vector using line search (52). The iterations are repeated until convergence.

D. Decomposition into block terms

In this section we develop an algorithm for decomposition into block terms given by (19) and (20). Given the estimate $\hat{\boldsymbol{Y}}$ we wish to compute the set of JxJ symmetric matrices $\boldsymbol{Q}_m$ given by (18) for $m = 1, \dots, M$ and the NxM matrix $\boldsymbol{C} = [c_{im}]$ by minimizing the following weighted least squares criterion

$$C_{bwls} = \sum_{j=1}^{J}\sum_{n=j}^{J}\sum_{i=1}^{N} \omega_i \left(\hat{\boldsymbol{Y}}(j,n,i) - \sum_{m=1}^{M}\sum_{r=1}^{R_m} k_{rm}f_{jrm}f_{nrm}c_{im}\right)^2. \tag{54}$$

where we set $\omega_i = \hat{p}_i$. We assume that the parameters M and $R = R_1 + \dots + R_M$ are known or estimated from $\hat{\boldsymbol{Y}}$ using (26) and (33). We propose the following three-step procedure for decomposition of $\hat{\boldsymbol{Y}}$ into block terms.

First, we observe that decomposition into block terms in (19) is equivalent to a decomposition into R rank-one terms, where the rank-one terms from the same block term have the same activity sequence vector $\boldsymbol{c}_m = \begin{bmatrix}c_{1m} & \dots & c_{Nm}\end{bmatrix}^T$ for $m = 1, \dots, M$. Hence, our first step is to compute a decomposition of $\boldsymbol{Y}$ into R rank-one terms. We impose appropriate nonnegativity and normalization constraints and use the algorithm described in Section III.B to compute

$$[\tilde{\boldsymbol{F}}, \tilde{\boldsymbol{C}}, \tilde{\boldsymbol{k}}] = \underset{\boldsymbol{F}\geq 0, \boldsymbol{1}^T\boldsymbol{F}=\boldsymbol{1}^T, \boldsymbol{C}\geq 0, \boldsymbol{1}^T\boldsymbol{C}=\boldsymbol{1}^T, \boldsymbol{k}}{\operatorname{argmin}} C_{wls}(\boldsymbol{F}, \boldsymbol{C}, \boldsymbol{k}). \tag{55}$$

where the matrices $\tilde{\boldsymbol{F}} = \begin{bmatrix}\tilde{\boldsymbol{f}}_1 & \dots & \tilde{\boldsymbol{f}}_R\end{bmatrix}$, $\tilde{\boldsymbol{C}} = \begin{bmatrix}\tilde{\boldsymbol{c}}_1 & \dots & \tilde{\boldsymbol{c}}_R\end{bmatrix}$ and $\tilde{\boldsymbol{k}} = \begin{bmatrix}\tilde{k}_1 & \dots & \tilde{k}_R\end{bmatrix}$ define rank-one

terms. In the second step we wish to group rank-one terms into block terms. Since we imposed nonnegativity and unit length constraints on the columns of $\tilde{\boldsymbol{C}}$, they should form M clusters, where each cluster defines one block term. We find these clusters by applying agglomerative hierarchical clustering [23] on the columns of $\tilde{\boldsymbol{C}}$. Initially, each vector belongs to a separate cluster. These clusters are then successively merged so that in each step a Euclidean distance based criterion measuring dispersion around cluster centroids is minimized. We continue this process until the number of clusters is reduced to M. The clustering algorithm finds the parameters $R_1, \ldots, R_M$ and the matrix of cluster centroids $\tilde{\boldsymbol{C}}^{(cl)} = \begin{bmatrix} \tilde{\boldsymbol{c}}_1^{(cl)} & \ldots & \tilde{\boldsymbol{c}}_M^{(cl)} \end{bmatrix}$, which will be used in the third step, where we seek parameter estimates

$$[\hat{\boldsymbol{F}_B}, \hat{\boldsymbol{C}}, \hat{\boldsymbol{k}_B}] = \underset{\substack{\boldsymbol{F}_B \geq 0, \mathbf{1}^T \boldsymbol{F}_B = \mathbf{1}^T, \\ \boldsymbol{C} = [c_{im} \in \{0,1\}], \boldsymbol{k}_B}}{\operatorname{argmin}} C_{bwls}(\boldsymbol{F}_B, \boldsymbol{C}, \boldsymbol{k}_B). \tag{56}$$

where $\boldsymbol{F}_B = \begin{bmatrix} \boldsymbol{F}_1 & \ldots & \boldsymbol{F}_M \end{bmatrix}$ and $\boldsymbol{k}_B = \begin{bmatrix} k_{11} & \ldots & k_{R_1,1} & \ldots & k_{1M} & \ldots & k_{R_M,M} \end{bmatrix}$ contain the parameters defined in (18). We solve (56) using an alternating least squares strategy, where we find an update for one set of parameters assuming that all other parameters are known. Starting form an initial point computed using results from the first two steps, we perform the following two steps in each iteration of the algorithm.

Update for $\hat{\boldsymbol{F}}_B$ and $\hat{\boldsymbol{k}}_B$. Assuming $\boldsymbol{C} = \hat{\boldsymbol{C}}$ we seek the estimates

$$[\hat{\boldsymbol{F}}_B, \hat{\boldsymbol{k}}_B] = \underset{\boldsymbol{F}_B \geq 0, \mathbf{1}^T \boldsymbol{F}_B = \mathbf{1}^T, \boldsymbol{k}_B}{\operatorname{argmin}} C_{bwls}(\boldsymbol{F}_B, \boldsymbol{C}, \boldsymbol{k}_B). \tag{57}$$

This problem can be solved using Newton's method with nonnegativity and unit norm constraints in the same way as described in Section III.C.

Update for $\hat{\boldsymbol{C}}$. We find this update assuming $\boldsymbol{F}_B = \hat{\boldsymbol{F}}_B$ and $\boldsymbol{k}_B = \hat{\boldsymbol{k}}_B$. Using $vecr$ operator the criterion function (54) can be written as

$$C_{bwls} = \sum_{i=1}^{N} \omega_i \| vecr(\hat{\boldsymbol{Y}}_i) - \sum_{r=1}^{M} vecr(\boldsymbol{Q}_r) c_{ir} \|_2^2 \tag{58}$$

where $\boldsymbol{Q}_r$ is computed from $\boldsymbol{F}_B$ and $\boldsymbol{k}_B$ using (18). Let us define $\boldsymbol{c}_i = \begin{bmatrix} c_{i1} & \ldots & c_{iM} \end{bmatrix}^T$ for $i = 1, \ldots, N$. From (58) we see that update for each $\hat{\boldsymbol{c}}_i$ can be computed separately. The update is found by solving

$$\min_{\boldsymbol{c}_i = [c_{im} \in \{0,1\}]} || vec(\hat{\boldsymbol{Y}}_i) - \begin{bmatrix} vec(\boldsymbol{Q}_1) & \ldots & vec(\boldsymbol{Q}_M) \end{bmatrix} \boldsymbol{c}_i ||_2^2 \tag{59}$$

which is done by evaluating criterion function in (59) for each of the 2^M possible values of $\boldsymbol{c}_i$ and picking the one that yields the lowest value of the criterion function [24]. The algorithm iteratively computes updates for $\hat{\boldsymbol{F}}_B$ and $\hat{\boldsymbol{k}}_B$, and $\hat{\boldsymbol{C}}$ until convergence. The matrix $\hat{\boldsymbol{C}}$ contains estimates of activity sequences over the temporal segments computed using the clustering algorithm as described in Section II.A. We still need to find an estimate of $\boldsymbol{B}$ defined in (3), which contains activity sequences over time intervals of duration T defined in (3). Using the definition of parameters $\boldsymbol{W} = [w_{ik}]$ and their estimates $\hat{\boldsymbol{W}} = [\hat{w}_{ik}]$ obtained using EM algorithm, we have $\hat{\boldsymbol{B}} = \hat{\boldsymbol{W}}^T \hat{\boldsymbol{C}}$.

IV. 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 1. Channels between the sensor and sources are transfer functions measured in ORBIT room in WINLAB for the setup shown in Figure 1 [25]. Thus, the real world propagation environment is faithfully reconstructed in the simulations.

Example 1. Both sources are transmitting DBPSK signals with Barker sequence spreading used in 802.11b systems [26]. The sources transmit in the same channel with equal power. The sensor observes one 20 MHz wide channel over the observation interval of 40 ms. Figure 2 shows power trace of the received signal at the sensor, where each point is the average power computed over the interval of T=10μs. We see 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 this example are -0.48 and 5.07 dB. Both sources transmit linearly modulated signals and hence, contribute rank-one components in the observed three-way array $\hat{\boldsymbol{Y}}$. Applying the proposed method with $R = M = 2$ recovers the trispectrum slice and activity sequence for each source, shown in Figures 3 and 4. Figure 3 shows magnitude of diagonal entries(kurtosis values) of the recovered trispectrum slices versus frequency. We see that the contributions from the two sources have been correctly recovered. The recovery of the two spectra in this example is possible due to the strong influence of the frequency selective channels between the sensor and the sources, which is evident in the recovered spectra in Figure 3. In frequency flat channels these two sources would contribute only one rank-one component in $\hat{\boldsymbol{Y}}$ since they use identical signals and hence, have proportional trispectrum slices(i. e., $\boldsymbol{Q}_1 = \alpha \boldsymbol{Q}_2$ in (12)). In that case, we would get from (26) and (33) $R = M = 1$. Applying our method for $R = M = 1$ without imposing $c_{im} \in \{0, 1\}$ would recover the common trispectrum slice and one temporal sequence describing the activity of both sources. We note that this decomposition is also a useful result of signal analysis. Resolving the contributions of the two sources may be possible by using multiple sensors which will be considered in our future work.

Example 2. One source is transmitting DBPSK signals with Barker sequence spreading and the other source is transmitting GFSK signal with frequency hopping used in Bluetooth systems [27]. Each Bluetooth packet is transmitted over one of 79 different channels, where each channel is approximately 1 MHz wide. The SNR values are -0.48(802.11b) and 6.76dB (Bluetooth). The spectrogram of the received signal computed

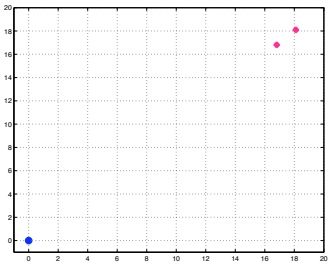

Fig. 1. Locations of the sensor(blue) and sources(red). Distances are in meters.

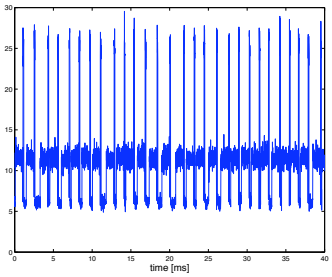

Fig. 2. Power trace of the received signal computed with $T = 10\mu s$

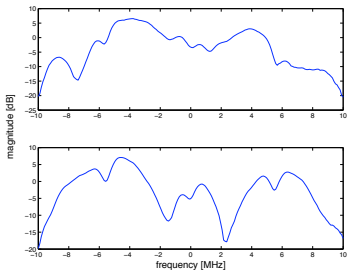

Fig. 3. Magnitude of diagonal entries(kurtosis values) of the recovered trispectrum slices

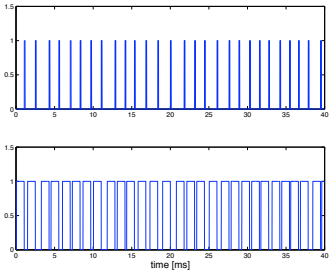

Fig. 4. Recovered activity sequences

with time resolution of T=10μs is shown in Figure 5. The sensor observes one 20 MHz wide channel over the 3 ms time interval during which each source transmits two packets. Both Bluetooth packets collide with 802.11b packets as can be seen in Figure 5. As already mentioned DBPSK signal with Barker sequence spreading is an example of linear modulation and hence, it contributes one rank-one term to the observed three-way array $\hat{\boldsymbol{Y}}$. The two Bluetooth packets are transmitted using GFSK signals at two different frequencies. These two signals have different spectra and hence, Bluetooth source contributes two spectra in $\hat{\boldsymbol{Y}}$. We conclude from this analysis that, in this example, $M = 3$. GFSK signal is an FM signal with Gaussian pulse shaping [28]. This signal is nonlinear. Hence, each GFSK signal contributes possibly multiple rank-one terms in $\hat{\boldsymbol{Y}}$. We applied our method for $M = 3$ and increasing number of rank-one terms starting with $R = 3$ and observed the computed spectra and activity sequences. The best results were obtained for $R = 6$. For $R < 6$ some meaningful rank-one terms were missed and for $R > 6$ some included rank-one terms were due to estimation errors in $\hat{\boldsymbol{Y}}$. The recovered $M = 3$ block terms have $R_1 = 3$(due to the first GFSK signal), $R_2 = 2$(due to the second GFSK signal), and $R_3 = 1$(due to the DBPSK signal) rank-one terms. The results are shown in Figures 6 and 7, where the graphs in the two top rows correspond to GFSK signals and the bottom graphs to DBPSK signals with Barker sequence spreading. The contributions of different signals have been correctly resolved despite the collisions between the packets. In this example the first GFSK signal is modeled using $R_1 = 3$ rank-one terms and the second GFSK signal is modeled using $R_2 = 2$ rank-one terms, which seems inconsistent. This effect can be intuitively explained as follows. Each GFSK signal is nonlinear and requires infinite number of rank-one terms for perfect representation. However, it can be well approximated using several most significant rank-one terms. The observed three-way array $\hat{\boldsymbol{Y}}$ consists of systematic variation modeled by six rank-one terms that form three block terms and error term $\boldsymbol{E}$. The systematic variation includes three rank-one terms from the first GFSK signal and two from the second GFSK signal because of significant difference in the received powers(about 4.8 dB) of these two signals, which is caused by frequency selective channel between the source and the sensor.

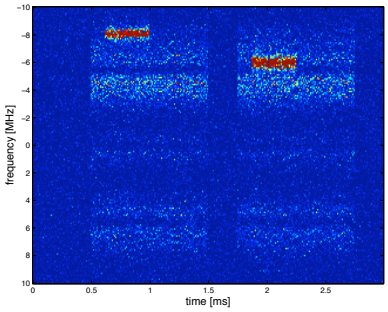

Fig. 5. Spectrogram of the received signal computed with $T = 10\mu s$

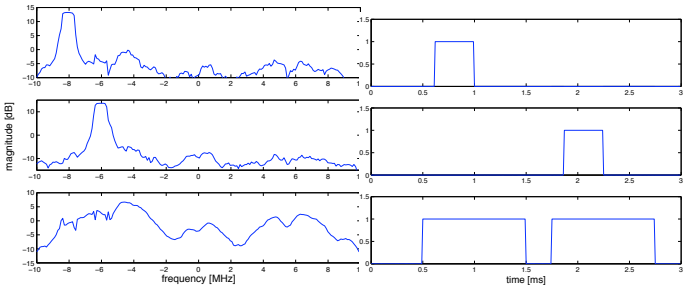

Fig. 6. Magnitude of diagonal entries(kurtosis values) of the recovered trispectrum slices

Fig. 7. Recovered activity sequences

Example 3. One source is transmitting DBPSK signals with Barker sequence spreading and the other is transmitting OFDM signal used in 802.11a/g systems [26]. The sensor observes one 20 MHz wide channel over the observation interval of $40\mu s$. The SNR values are -0.48(DBPSK) and 4.84 dB(OFDM). The transmitted packets from the two sources are interleaved in time and there are no collisions as can be seen from Figure 8. Obviously, in this example $M = 2$. We know that DBPSK signal with Barker sequence spreading contributes one rank-one term to the observed three-way array $\hat{\boldsymbol{Y}}$. OFDM signal is a superposition of 52 linearly modulated carriers. Hence, OFDM signal is nonlinear and its trispectrum slice consists of 52 rank-one terms. Now, we should apply our method for $M = 2$ and $R = 53$. Since this computation requires a lot of memory and is very slow we applied our method for $M = 2$ and $R = 25$. The two recovered block terms here have $R_1 = 1$(due to the DBPSK signal) and

$R_2 = 24$(due to the OFDM signal) rank-one terms. The results are shown in Figures 9 and 10, where the graphs in the top rows correspond to the DBPSK signal and the bottom graphs to the OFDM signal. Again, we observe significant effects of the frequency selective channels on the recovered spectra. Although, the OFDM signal was modeled using only $R_2 = 24$ rank-one terms, the algorithm still captured the most significant signal components. This example illustrates potential computational difficulties in application of the proposed method. But it also illustrates how fourth order spectrum reveals the difference between single carrier(DBPSK) and multicarrier(OFDM) modulations, which cannot be seen from second order spectrum.

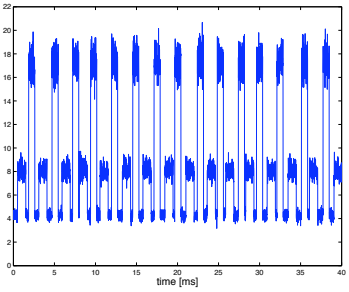

Fig. 8. Power trace of the received signal computed with $T = 10\mu s$

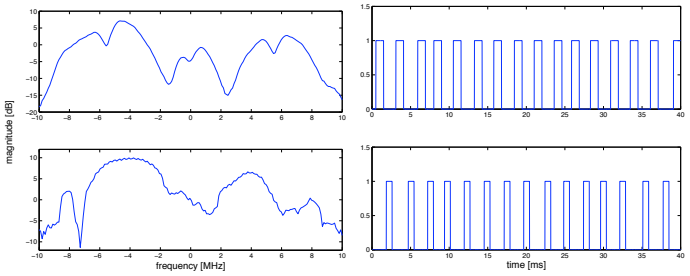

Fig. 9. Magnitude of diagonal entries(kurtosis values) of the recovered trispectrum slices

Fig. 10. Recovered activity sequences

V. Conclusions and Future Work

This work shows that fourth order spectrum could be a very useful tool for analysis of packet based radio signals. Fourth order spectrum contains important information about the received signal which is not contained in the second order spectrum and thus, it enables design of qualitatively different algorithms. Currently, we are developing an extension of the method that uses multiple sensors. A more detailed performance analysis of these algorithms will be presented in our future publications.

References

[1] D. Cabric, "Cognitive radios: System design perspective," Ph.D. dissertation, Univ. of California, Berkeley, Nov 2007.

[2] 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.

[3] C. M. Spooner, "Multi-resolution white-space detection for cognitive radio," *MILCOM 2007*., pp. 1–9, 29-31 Oct. 2007.

[4] D. Noguet. (2009, April) "Sensing techniques for cognitive radio-State of the art and trends"(White paper). [Online]. Available: http://www.scc41.org/

[5] 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.

[6] S. M. Kay, *Fundamental of Statistical Signal Processing: Detection Theory*. Upper Saddle River, NJ: Prentice-Hall, 1998.

[7] P. Stoica and R. Moses, *Introduction to Spectral Analysis*. Upper Saddle River, NJ: Prentice-Hall, 1997.

[8] Fraley C. and Raftery A.E., "Model-Based Clustering, Discriminant Analysis, and Density Estimation," *Journal of the American Statistical Association*, vol. 97, pp. 611–631(21), 1 June 2002.

[9] 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.

[10] C. L. Nikias and A. P. Petropulu, *Higher-order Spectra Analysis: A Nonlinear Signal Processing Framework*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] A.-J. van der Veen, "Joint diagonalization via subspace fitting techniques," *ICASSP 2001*, vol. 5, pp. 2773–2776, 2001.

[19] P. Paatero, "A weighted non-negative least squares algorithm for three-way PARAFAC factor analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 38, October 1997.

[20] ——, "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.

[21] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2004.

[22] P. E. Gill, W. Murray, and M. H. Wright, *Practical optimization*. London: Academic Press, 1981, 1981.

[23] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.

[24] 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.

[25] 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, Sept. 30 2007-Oct. 3 2007.

[26] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11, 1999.

[27] J. C. Hartsen, "Bluetooth radio system," *IEEE Personal Communications*, vol. 7, no. 1, pp. 28–36, Feb. 2000.

[28] T. Rappaport, *Wireless Communications: Principles and Practice*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.