Polyphonic Sound Event Tracking using Linear Dynamical Systems

Emmanouil Benetos, *Member, IEEE,* Grégoire Lafay, Mathieu Lagrange, and Mark D. Plumbley, *Fellow, IEEE*

***Abstract*—In this paper, a system for polyphonic sound event detection and tracking is proposed, based on spectrogram factorisation techniques and state space models. The system extends probabilistic latent component analysis (PLCA) and is modelled around a 4-dimensional spectral template dictionary of frequency, sound event class, exemplar index, and sound state. In order to jointly track multiple overlapping sound events over time, the integration of linear dynamical systems (LDS) within the PLCA inference is proposed. The system assumes that the PLCA sound event activation is the (noisy) observation in an LDS, with the latent states corresponding to the true event activations. LDS training is achieved using fully observed data, making use of ground truth-informed event activations produced by the PLCA-based model. Several LDS variants are evaluated, using polyphonic datasets of office sounds generated from an acoustic scene simulator, as well as real and synthesized monophonic datasets for comparative purposes. Results show that the integration of LDS tracking within PLCA leads to an improvement of +8.5-10.5% in terms of frame-based F-measure as compared to the use of the PLCA model alone. In addition, the proposed system outperforms several state-of-the-art methods for the task of polyphonic sound event detection.**

***Index Terms*—Sound event detection, linear dynamical systems, probabilistic latent component analysis, sound scene analysis.**

I. Introduction

Sound event detection (SED), also called acoustic event detection, is a central topic in the emerging field of sound scene analysis. The main goal of SED is to label temporal regions within an audio recording, resulting in a symbolic description with start and end times, as well as labels[1] for each instance of a specific event type [1]. Applications for sound event detection are numerous, including but not limited to security and surveillance, urban planning, smart homes, acoustic ecology, and organisation/navigation of sound archives [1] [2] [3] [4].

The majority of research in sound event detection is on detecting one acoustic event at a given time segment, which is referred to as *monophonic* sound event detection, or as detection of non-overlapping acoustic events. Methods that address the problem of detecting overlapping events from audio (also called *polyphonic* sound event detection) include the work by Heittola et al. [3] on using a context-dependent Hidden Markov Models (HMMs) with multiple path decoding. Gemmeke et al. [5] proposed the use of vectorized time-frequency patches of pre-extracted isolated events within the context of non-negative matrix factorization (NMF). Dennis et al. [4] proposed a method for detecting overlapping sound events using local spectrogram features and a Generalised Hough Transform voting system. As part of the 2013 IEEE AASP challenge on Detection and Classification of Acoustic Scenes and Events (DCASE 2013) [6], a baseline system was created using NMF with beta-divergence. Also as part of the DCASE 2013 challenge, Vuegen et al. [7] proposed a system based on Gaussian mixture models (GMMs), with Mel-frequency cepstral coefficients (MFCCs) as input features. More recently, Mesaros et al. [8] proposed the use of coupled NMF for sound event detection, which bypasses the supervised construction of class models. Finally, Komatsu et al. [9] perform sound event detection using NMF with mixtures of local dictionaries and activation aggregation.

With respect to the use of connectionist approaches to the problem of sound event detection, Cakir et al. [10] used multilabel deep neural networks with spectral features as inputs. This work was continued in [11], which applied bi-directional long short term memory recurrent neural networks (BLSTM RNNs) for the same task. It is worth noting that the methods of [10] [11] were only applied on proprietary data.

Given that SED systems have to produce a series of events identified by a start and end time, modelling temporal dynamics is crucial. Currently, most systems either produce a frame-based posteriogram or event activation, which is subsequently thresholded [5] [8] [12], or they incorporate temporal information by computationally expensive convolutional formulations [13] [14] or vectorized time-frequency patches [5]. A subset of sound event detection systems incorporate temporal constraints for polyphonic SED in the form of HMMs. Since HMMs only support one discrete latent state at a given time instant, polyphony is supported through multiple Viterbi passes [3] or through multiple HMMs [12]. The preliminary system of [12] forms the basis of this current work; it used a spectrogram factorisation-based sound event detection system which imposed temporal constraints on the appearance of each *sound state* of an event in the form of independent event-wise HMMs. While extensions of HMMs, such as factorial HMMs [15], are able to support several concurrent Markov chains and could be

E. Benetos is with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, U.K. (e-mail: emmanouil.benetos@qmul.ac.uk).

G. Lafay and M. Lagrange are with IRCCYN, CNRS, École Centrale de Nantes, 44321 Nantes Cedex 3, France (e-mail: gregoire.lafay@irccyn.ec-nantes.fr, mathieu.lagrange@cnrs.fr).

M. D. Plumbley is with the Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford GU2 7XH, U.K. (e-mail: m.plumbley@surrey.ac.uk).

EB is supported by a UK Royal Academy of Engineering Research Fellowship (grant no. RF/128). MDP is partly supported by UK Engineering and Physical Sciences Research Council (EPSRC) Grants EP/N014111/1, EP/L027119/1, and EP/L027119/2.

[1] It should be noted that the concept of SED also includes the identification/classification of sound events, in addition to detecting their start and end times.

used in polyphonic SED, they are in practice computationally prohibitive in the case of unconstrained polyphony.

In the present work, we propose a system for polyphonic sound event detection based on spectrogram factorisation approaches, which uses linear dynamical systems (LDS - see subsection II-B) for tracking multiple concurrent events across time. LDS can be viewed as a generalisation of HMMs, where the latent space in LDS is continuous and multi-dimensional. The spectrogram factorisation model is based on probabilistic latent component analysis (PLCA - see subsection II-A) and decomposes an input audio spectrogram into a series of probability distributions for event activations, exemplar contributions, and sound state activations. To the authors' knowledge, this is the first time that LDS have been applied to the field of sound scene analysis, and this is the first attempt on jointly tracking multiple sound events instead of using event independence assumptions, such as was done in the HMM-based system of [12].

The proposed polyphonic sound event tracking method uses the event activation output of the PLCA-based spectrogram factorisation model as the (noisy) observation of an LDS, where the latent states correspond to the 'true' event activations. Thus, the LDS can provide a mapping between a noisy system output and a 'clean' polyphonic detection. LDS parameters are learned at a training stage using fully observed data, which correspond to pairs of sound event detection outputs and ground truth-informed outputs. The proposed method is trained on datasets from the DCASE 2013 challenge [1] and tested on several polyphonic datasets of office sounds, under variable noise and event density conditions. Results show that the proposed LDS-based event tracking method can provide a significant and consistent improvement over the use of the event activation output directly. At the same time, the proposed LDS-based event tracking is robust to changes in acoustic and recording conditions, and the resulting system is able to outperform several state-of-the-art polyphonic sound event detection approaches for the same task.

The outline of this paper is as follows. Section II presents background information on the standard PLCA and LDS models. The proposed system is described in Section III, including motivation for this work, pre-processing, the extended PLCA model, and LDS-based sound event tracking. Evaluation, including a description of the train/test datasets, evaluation metrics, and experimental results, is presented in Section IV. Finally, conclusions are drawn and future directions are discussed in Section V.

II. Background

A. Probabilistic Latent Component Analysis

Probabilistic latent component analysis (PLCA) is a spectrogram factorisation technique proposed in [16]. It can be viewed as a probabilistic extension of non-negative matrix factorization (NMF) [17] using the Kullback-Leibler cost function. PLCA can also offer a convenient way to incorporate priors over the model parameters and control the resulting decomposition [18] [19]. In PLCA, the input spectrogram $V_{f,t}$ is modeled as the histogram of the draw of independent random variables $\{f, t\}$ which are distributed according to the bivariate probability distribution $P(f, t)$, where f denotes the frequency index and t the time index. The PLCA model expresses $P(f, t)$ as a mixture of latent factors.

There are two ways of modeling $P(f, t)$, using symmetric or asymmetric factorisations. The asymmetric model, which is popularly known as probabilistic latent semantic analysis (PLSA) in the literature of topic modelling [18], decomposes $P(f, t)$ as a product of a spectral basis matrix (also called spectral template matrix) and a component activation matrix:

$$P(f,t) = P(t) \sum_{d} P(f|d) P(d|t) \tag{1}$$

where d is the component index, $P(t)$ is the l_1 norm for the t-th spectrogram frame (a known quantity), $P(f|d)$ is the spectral template that corresponds to the d-th component, and $P(d|t)$ is the activation of the d-th component over t. Using the same variables as in (1), the symmetric model decomposes $P(f, t)$ as:

$$P(f,t) = \sum_{d} P(d) P(f|d) P(t|d) \tag{2}$$

where $P(d)$ corresponds to the component prior and $P(t|d)$ contains the latent marginal distribution across time t relating to component d.

In order to estimate $P(f|d)$ and $P(d|t)$ in the asymmetric model or $P(d)$, $P(f|d)$, and $P(t|d)$ in the symmetric model, iterative update rules are applied using the Expectation-Maximization (EM) algorithm [20]. The derivation of the EM algorithm for PLCA can be found in [21]. The update rules are guaranteed to converge to a local minimum. In the context of audio signal analysis, the components (or latent factors) d typically refer to the constituent elements of a spectrogram, such as acoustic events or sound sources.

B. Linear Dynamical Systems

Sequential data can be represented using a Markov chain of latent variables, with each observation conditioned on the state of the corresponding latent variable [22]. If the latent variables are discrete, we obtain a hidden Markov model (HMM) [23]. *State space models* (SSMs) are generalisations of HMMs, where the hidden states are continuous [15]. A special case of an SSM is where the latent and observed variables are multivariate Gaussian distributions whose means are linear functions of their parent states. This model is called a *linear-Gaussian SSM* (LG-SSM) or a *linear dynamical system* (LDS) [15] [22]. Historically, LDS were developed independently of HMMs, and are widely known in the signal processing community as *Kalman filters* [24]; the relationship between HMMs and Kalman filters has recently been noted in the context of machine learning [22] [15]. A graphical representation of an LDS can be seen in Fig. 1. The representation is equivalent to that of an HMM, with the exception that in an HMM the latent variable $\mathbf{z}_t$ is discrete and one-dimensional.

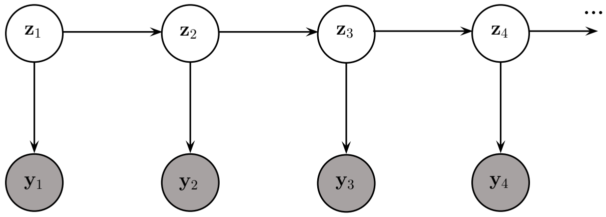


Fig. 1. Graphical representation of an LDS.

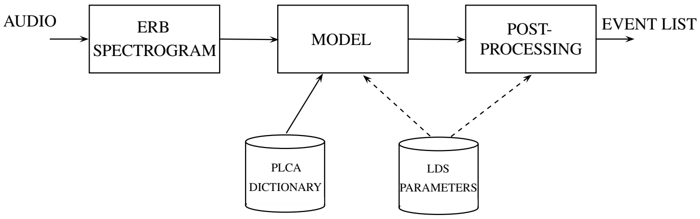


Fig. 2. Proposed system diagram.

An LDS can be formulated as:

$$\begin{aligned}\mathbf{z}_t &= \mathbf{A}_t\mathbf{z}_{t-1} + \epsilon_t \\ \mathbf{y}_t &= \mathbf{B}_t\mathbf{z}_t + \delta_t \\ \epsilon_t &\sim \mathcal{N}(0, \mathbf{Q}_t) \\ \delta_t &\sim \mathcal{N}(0, \mathbf{R}_t)\end{aligned} \tag{3}$$

where $\mathbf{z}_t$ is the hidden state, $\mathbf{A}_t$ is the transition model, ϵ_t is Gaussian system noise (with covariance $\mathbf{Q}_t$), $\mathbf{y}_t$ is the observation, $\mathbf{B}_t$ is the observation model, and δ_t is Gaussian observation noise (with covariance $\mathbf{R}_t$). In the following, the LDS will be assumed to be stationary, and the subscript t will be omitted from $\mathbf{A}_t, \mathbf{B}_t, \mathbf{Q}_t, \mathbf{R}_t$. A useful property of LDS is that they support exact inference, which is expressed by the *Kalman filter* equations for estimating the online posterior $P(\mathbf{z}_t|\mathbf{y}_{1:t})$, and the *Kalman smoother* equations for estimating the offline posterior $P(\mathbf{z}_t|\mathbf{y}_{1:T})$ [15] (where T is the length of the sequence).

Applications of LDS are numerous (see [15] for an overview), although to the authors' knowledge LDS have not yet been applied in the emerging field of sound scene analysis. Recently, two NMF-based models were proposed for speech denoising and separation tasks, which incorporated temporal constraints similar to those of an LDS. In [25], an extension of NMF was proposed which supported Markovian dynamics: the observation model operates similarly to standard NMF, while the latent dynamics capture statistical dependencies between time frames similarly to LDS. In [26], a dynamic NMF model is proposed, where the observation model is similar to NMF/PLCA and follows a multinomial distribution, and the encoding matrix dynamics are formulated using an autoregressive model.

III. Proposed Method

A. Motivation and System Overview

The overall aim of the proposed work is the creation of a system for polyphonic sound event detection that also supports joint tracking of sound events over time. In this paper, we aim to express a sound event as a linear combination of exemplars for a specific event class, where each exemplar consists of a collection of *sound state* spectral templates (a sound state refers to an instance in the temporal evolution of a specific exemplar). Thus, the model is based on a 4-dimensional dictionary of frequency, sound event class, exemplar index, and sound state index. It should be noted that the proposed PLCA-based model is expressed as a mixture of latent components corresponding to sound events, and thus cannot jointly model multiple concurrent sound events. The model can however infer the presence of concurrent sound events by calculating the posterior probability of each sound event over all possible events (see Sec. III-C).

In addition, the proposed model aims to jointly track multiple concurrent sound events over time using linear dynamical systems, and improve upon the PLCA-based estimation of the sound event activation by incorporating LDS-based sound event tracking. In contrast with HMMs, which support a one-dimensional discrete latent variable, LDS support a multi-dimensional and continuous latent variable space. Thus, the LDS can provide a mapping between an observed combination of sound events and a 'true' combination of sound events.

With respect to previous work on combining NMF with LDS: while the methods of [25] [26] are able to provide a component activation matrix that is able to evolve smoothly over time, in the present work we are primarily interested in using the LDS in a supervised scenario, to provide a mapping between the observed 'noisy' output of an event detection system and the latent 'true' sound event output, which is not possible using the aforementioned methods.

A diagram for the proposed system is shown in Fig. 2. The proposed sound event detection system takes as input an audio recording and computes a time-frequency representation, which is subsequently used as input to the proposed model. The model uses a pre-extracted dictionary of sound event spectral templates used in the PLCA-based model. Sound event tracking using LDS can take place within the PLCA inference (dashed arrow from "LDS parameters" to "Model" in Fig. 2) or can take place as a post-processing step (dashed arrow from "LDS parameters" to "Post-processing"). The model output is finally converted into a list of sound events identified by a start time, end time, and sound event class.

B. Preprocessing

The proposed model first computes a time-frequency representation of an audio recording, denoted as $V_{f,t}$, where $f \in \{1, \ldots, F\}$ is the frequency index and $t \in \{1, \ldots, T\}$ is the time index. Here, $V_{f,t}$ is created by subsampling the input signal to 22.05kHz and processing it with an equivalent rectangular bandwidth (ERB) filterbank [27], following the method of [28]. This auditory-motivated and relatively compact filterbank uses 250 filters that consist of sinusoidally modulated Hanning windows, linearly spaced between 5Hz and 10.8kHz on the ERB scale. Each subband is partitioned into disjoint 23ms time frames, and the root mean square of the filterbank output is computed for each frame.

C. PLCA model

The proposed PLCA-based model takes as input the ERB spectrogram $V_{f,t}$ and approximates it as a bivariate probability distribution $P(f,t)$. The model decomposes the approximated spectrogram $P(f,t)$ into a dictionary of spectral templates per event class s, exemplar index c, and sound state q, as well as probability distributions for event activations, exemplar contributions per class, and sound state activations per event class. The model is formulated as:

$$P(f,t) = P(t) \sum_{q,c,s} P(f|q,c,s)P(s|t)P(c|s,t)P(q|s,t) \quad (4)$$

where $s \in \{1,\ldots,S\}$ denotes the sound event class, $c \in \{1,\ldots,C\}$ denotes the exemplar index, and $q \in \{1,\ldots,Q\}$ the sound state index. $P(t)$ is defined as $\sum_f V_{f,t}$, which is a known quantity, corresponding to the sum of all frequency bins in the ERB spectrogram for each time frame t. Dictionary $P(f|q,c,s)$ is a 4-dimensional tensor that contains the spectral templates for sound event s, exemplar c and sound state q. $P(s|t)$ is the time-varying event activation. $P(c|s,t)$ denotes the time-varying exemplar contribution for producing a specific event s at a given time frame t. Finally, $P(q|s,t)$ is the sound state activation per event class s, across time t.

In the model of (4), spectral templates $P(f|q,c,s)$ are normalised with respect to f as to sum to one, in order to be regarded as probabilities. $P(s|t)$, $P(c|s,t)$, and $P(q|s,t)$ are similarly normalised with respect to s, c, and q, respectively. Conversely, $P(f,t)$ and $P(t)$ are not normalised since they carry information on the energy of the spectrogram. However this does not affect inference since $P(t)$ and $P(f,t)$ are cancelled out through the partition functions.

The unknown model parameters $P(s|t)$, $P(c|s,t)$, and $P(q|s,t)$ can be estimated using iterative update rules such as the Expectation-Maximization (EM) algorithm [20]. For the *E-step*, the following posterior is computed:

$$P(q,c,s|f,t) = \frac{P(f|q,c,s)P(s|t)P(c|s,t)P(q|s,t)}{\sum_{q,c,s} P(f|q,c,s)P(s|t)P(c|s,t)P(q|s,t)}. \quad (5)$$

For the *M-step*, $P(s|t)$, $P(c|s,t)$ and $P(q|s,t)$ are updated using the posterior of (5):

$$P(s|t) = \frac{\sum_{q,c,f} P(q,c,s|f,t)V_{f,t}}{\sum_{s,q,c,f} P(q,c,s|f,t)V_{f,t}} \quad (6)$$

$$P(c|s,t) = \frac{\sum_{q,f} P(q,c,s|f,t)V_{f,t}}{\sum_{c,q,f} P(q,c,s|f,t)V_{f,t}} \quad (7)$$

$$P(q|s,t) = \frac{\sum_{c,f} P(q,c,s|f,t)V_{f,t}}{\sum_{c,q,f} P(q,c,s|f,t)V_{f,t}}. \quad (8)$$

The model of (4) can be further constrained by enforcing sparsity to certain unknown model parameters. Since for the sound event detection problem only a few sound event classes are expected to be active at a given time frame, sparsity can be imposed on the event activation $P(s|t)$. Likewise, an active sound event at a given time frame is expected to be produced by a limited number of exemplars, so sparsity can also be enforced on $P(c|s,t)$. Here, the sparsity constraints are achieved in a similar way to the method of [29], by modifying the update equations (6) and (7) to give:

$$P(s|t) = \frac{\left(\sum_{q,c,f} P(q,c,s|f,t)V_{f,t}\right)^{\kappa}}{\sum_s \left(\sum_{q,c,f} P(q,c,s|f,t)V_{f,t}\right)^{\kappa}} \quad (9)$$

$$P(c|s,t) = \frac{\left(\sum_{q,f} P(q,c,s|f,t)V_{f,t}\right)^{\lambda}}{\sum_c \left(\sum_{q,f} P(q,c,s|f,t)V_{f,t}\right)^{\lambda}}. \quad (10)$$

By setting $\kappa, \lambda > 1$ (typical values are between 1.1-1.5), the entropy in $P(s|t)$ and $P(c|s,t)$ is lowered and sparsity is promoted [29].

No update rule for the sound state templates $P(f|q,c,s)$ is included, since they are pre-extracted and considered fixed (see subsection IV-A on dictionary creation). The unknown parameters $P(s|t)$, $P(c|s,t)$ and $P(q|s,t)$ are initialised[2] in the EM updates with random values between 0 and 1. Eqs. (5) and (8)-(10) are iterated until convergence: in our experiments, we found 30 iterations to be sufficient.

The output of the PLCA model is a 2-dimensional non-binary representation of event activations over time, given by $P(s,t) = P(t)P(s|t)$ (with dimensions $S \times T$). Essentially, the output is created by calculating the posterior probability of each event over all possible events, i.e. $P(s=1|t)$, $P(s=2|t)$, ... , $P(s=S|t)$, weighted by energy of the ERB spectrogram.

D. LDS Learning

The PLCA model output $P(s,t)$ contains the non-binary activation of overlapping sound events s over time t. However the model of (4) does not incorporate any temporal constraints, and thus can lead to a temporally fragmented output. Here, we propose the use of LDS to perform polyphonic event tracking: to do this, we assume that the event activation $P(s,t)$ is a 'noisy' observation $\mathbf{y}_t$ in an LDS, for which the latent states $\mathbf{z}_t$ correspond to our desired output.

LDS learning, i.e. estimating the parameters $\mathbf{A}, \mathbf{B}, \mathbf{Q}$, and $\mathbf{R}$, if there is only access to observations, can be achieved using the EM algorithm [15], in a similar way to the HMM Baum-Welch algorithm [23]. However in our case we also have access to the hidden state sequences $\mathbf{z}_t$ which correspond to the 'true' event detection outputs, which can be used to perform LDS learning with *fully observed data* [15]. Obtaining the hidden state sequences is achieved by constraining the event activation in the PLCA model of (4) using event ground truth annotations. By initialising $P(s|t)$ in the EM updates with a binary mask that corresponds to the ground truth annotations, the resulting output (denoted as $P'(s,t)$) only has nonzero activations in the time instants and classes corresponding to ground truth events. An example ground truth annotation along with a ground truth-informed event detection output used for training the LDS can be seen in Fig. 3.

[2] As shown in [30], the accuracy of the model depends on the initialisation of unknown parameters. Experiments with multiple runs of the PLCA model with different random initialisations are shown in Section IV-E.

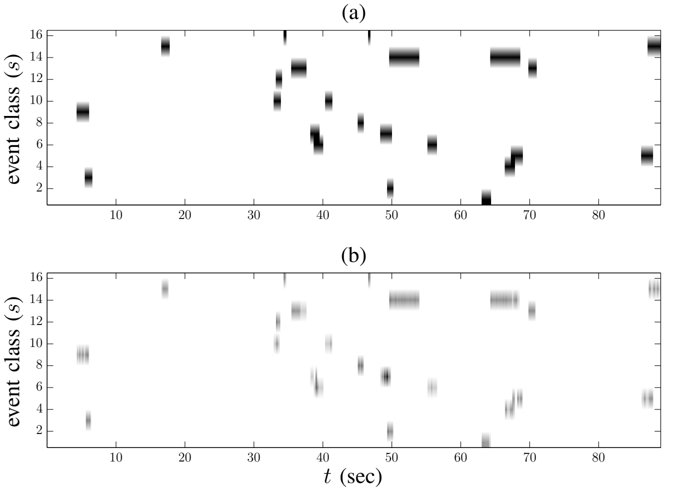


Fig. 3. (a) Sound event ground truth annotation for sound recording no.3 from the DCASE 2013 Challenge - OS development set [1]. (b) The corresponding ground truth-informed sound event detection output $P'(s,t)$. Sound classes $s \in \{1,\ldots,16\}$ are described in subsection IV-A.

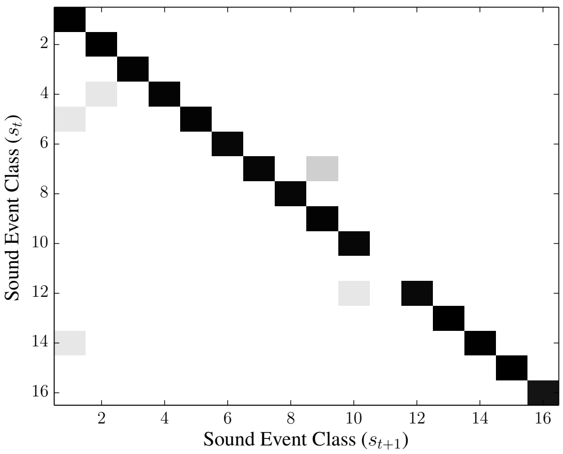


Fig. 4. The LDS transition matrix $\mathbf{A}$ trained on sequences of office sounds. Sound class indices 1-16 are listed in subsection IV-A.

Given fully observed data, the LDS model parameters $\mathbf{A}$ and $\mathbf{B}$ can be estimated by solving least squares problems for $\mathbf{z}_{t-1} \rightarrow \mathbf{z}_t$ and $\mathbf{z}_t \rightarrow \mathbf{y}_t$, respectively [15]:

$$\begin{aligned} J(\mathbf{A}) &= \sum_t (\mathbf{z}_t - \mathbf{A}\mathbf{z}_{t-1})^T(\mathbf{z}_t - \mathbf{A}\mathbf{z}_{t-1}) \\ J(\mathbf{B}) &= \sum_t (\mathbf{y}_t - \mathbf{B}\mathbf{z}_t)^T(\mathbf{y}_t - \mathbf{B}\mathbf{z}_t) \end{aligned} \tag{11}$$

where $(\cdot)^T$ denotes vector transpose, $\mathbf{y}_\tau = P(s,t=\tau)$ and $\mathbf{z}_\tau = P'(s,t=\tau)$. Without loss of generality [15], the system and observation noise covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ are here assumed to be diagonal in the form of $\mathbf{Q} = \alpha\mathbf{I}$ and $\mathbf{R} = \beta\mathbf{I}$, with scaling parameters $\alpha, \beta \in \mathbb{R}$ estimated from training data (see subsection IV-A for a discussion on training data). As an example, the transition matrix $\mathbf{A}$ estimated for the proposed sound event detection system is shown in Fig. 4 (see subsection IV-A for the training data used). Note that the main diagonal is strong, which favours tracking events over time, apart from event $s = 11$, which corresponds to a 'page turn' event class which is not present in the training data.

So far, we have assumed that the latent variable space in the LDS includes a one-to-one correspondence with the observed variables, where each latent variable corresponds to a sound event class. We also investigate an LDS variant where the latent variable space also includes 'velocity' values $\dot{\mathbf{z}}_t$ for each event class, signifying the difference in amplitude values in the event activation matrix $P(s,t)$ across adjacent time frames. This formulation is inspired by the *random accelerations model* used in object tracking using Kalman filters [15]. Using this approach, the latent variable space is now defined as:

$$\mathbf{z}_t = \begin{pmatrix} z_{1t} & \cdots & z_{St} & \dot{z}_{1t} & \cdots & \dot{z}_{St} \end{pmatrix} \tag{12}$$

where $z_{\varsigma,\tau} = P'(s=\varsigma, t=\tau)$ and $\dot{z}_{\varsigma,\tau} = P'(s=\varsigma, t=\tau) - P'(s=\varsigma, t=\tau-1)$.

E. LDS Inference & Postprocessing

LDS inference refers to estimating the model posterior $P(\mathbf{z}_t|\mathbf{y}_{1:t})$ in the online case or $P(\mathbf{z}_t|\mathbf{y}_{1:T})$ in the offline case, where $t \in \{1,\ldots,T\}$. Estimating the aforementioned posteriors can be achieved through the Kalman filter and Kalman smoother equations, respectively [24] [22]. The LDS inference process, which is similar to the HMM forward-backward algorithm [23], is omitted here for brevity. In the online case (i.e. having access only to past samples), the posterior is represented as: $P(\mathbf{z}_t|\mathbf{y}_{1:t}) = \mathcal{N}(\mathbf{z}_t|\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$, and the output of the online LDS-based sound event tracking process is the LDS posterior mean $\boldsymbol{\mu}_t$ (or the first half of the latent variables corresponding to $\boldsymbol{\mu}_t$ in the case of the random accelerations model). In the offline case (i.e. having access to both past and future samples), the LDS posterior is represented as $P(\mathbf{z}_t|\mathbf{y}_{1:T}) = \mathcal{N}(\mathbf{z}_t|\boldsymbol{\mu}_{t|T}, \boldsymbol{\Sigma}_{t|T})$ and the output of the sound event tracking process is the LDS posterior mean $\boldsymbol{\mu}_{t|T}$.

In this work, the aforementioned LDS-based event tracking process can either be applied as a post-processing step or can be integrated in the PLCA update equations. For the former, the PLCA model output $P(s,t)$ is post-processed using an LDS and results in the 'clean' output, which is the LDS posterior mean $\boldsymbol{\mu}_t$ or $\boldsymbol{\mu}_{t|T}$ (for the online and offline case, respectively).

A second use of the LDS sound event tracking process is to integrate it during PLCA inference, in the form of a Dirichlet prior [19]. Following the procedure of [19], we define the Dirichlet hyperparameter for the 'clean' event activation as:

$$\phi(s|t) \propto \boldsymbol{\mu}_t \tag{13}$$

Subsequently, we modify the update rule for the event activation as to include a weighted component with the LDS-based sound event tracking:

$$P(s|t) \propto (w-1) \cdot \left(\sum_{q,c,f} P(q,c,s|f,t) V_{f,t} \right)^{\kappa} + w \cdot \phi(s|t) \tag{14}$$

where $w \in \{0,1\}$ is a weight parameter indicating how much the prior should be imposed. The complete algorithm that uses the PLCA model with LDS integration can be seen in Algorithm 1. Although convergence is not guaranteed, it

Algorithm 1 PLCA-LDS integration

```
Require: $V_{f,t}$, $P(q,c,s|f,t)$, $\mathbf{A}$, $\mathbf{C}$, $\mathbf{R}$, $\mathbf{Q}$, $iter$
    1) Initialise $P(s|t)$, $P(c|s,t)$, $P(q|s,t)$
    2) Compute $P(t) = \sum_f V_{f,t}$
    3) For $i$=1:$iter$
        a) Compute $P(q,c,s|f,t)$ from (5)
        b) Compute $P(q|s,t)$ from (8)
        c) Compute $P(c|s,t)$ from (10)
        d) Compute $\boldsymbol{\mu}_t$ or $\boldsymbol{\mu}_{t|T}$ using the Kalman filter/smoother equations [24] [22]
        e) Compute $P(s|t)$ from (14)
```

is observed in practice, in terms of a constantly decreasing Kullback-Leibler divergence between the original and approximated spectrogram. It should be noted that the LDS does not impose any constraints on the inputs being non-negative or summing to one (although in practice the LDS posterior values are in the same range with $P(s,t)$). Thus, in order to ensure that the estimated event posterior $P(s|t)$ remains non-negative in (14), only the non-negative values of $\boldsymbol{\mu}_t$ (or $\boldsymbol{\mu}_{t|T}$) are kept. Normalisation then takes place to both weighted components of (14) as part of the equation's partition function.

The output of the LDS-based post-processing step or the PLCA-LDS integration of (14) is a smooth non-binary sound event activation, which needs to be converted into a list of detected events per time frame. In this work, the LDS output is binarised by performing class-specific thresholding (each sound event class is thresholded using (non-negative) value θ_s, estimated from a training set; see subsection IV-A). Finally, detected events with a small duration (here, detected events shorter than 60ms) are removed.

It should be noted that the value of the LDS posterior is dependent on the values of $P(s,t)$, which is expressed by the sound event posterior $P(s|t)$ weighted by the ERB spectrogram energy $P(t)$. As the number of concurrent sound events increase, the sound event posterior will always decrease. However, since an increased number of concurrent sound events will also lead to an increase in the energy of the ERB spectrogram, the values of $P(s,t)$ are essentially not affected by an increasing number of concurrent events. Thus, in practice the number of concurrent sound events does not cause any difficulties in finding suitable thresholds θ_s.

An example event detection output is shown in Fig. 5, comparing the output of a PLCA-only model, a PLCA model with HMM constraints on the sound state activation [12], PLCA with LDS postprocessing, and PLCA-LDS integration. When comparing the aforementioned outputs with the ground truth of Fig. 3, the LDS postprocessing (Fig. 5c) and LDS integration (Fig. 5d) are able to detect instances of class 14 'printer' in sec. 50-70 that were not detected in the PLCA-only and PLCA-HMM models. However LDS postprocessing (Fig. 5c) also introduces false alarms around sec. 50 which are not present in the PLCA-LDS integration output (Fig. 5d).

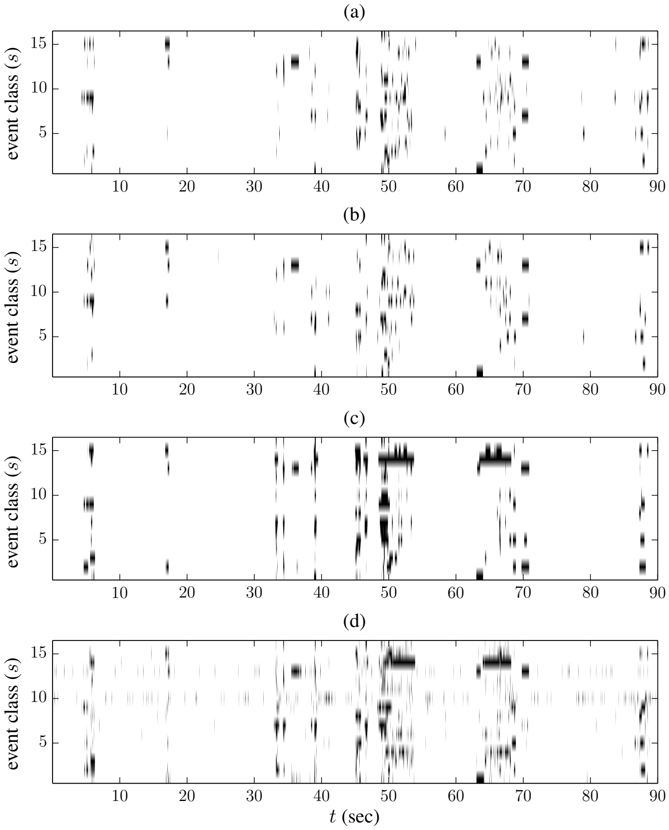


Fig. 5. Binary sound event detection outputs of the recording shown in Fig. 3. (a) Using the PLCA-only model, reaching a frame-based F-measure of 32.2%. (b) Using the PLCA model with HMM constraints on the sound state activation [12], reaching and F-measure of 37.3%. (c) Using the PLCA model with LDS postprocessing, reaching an F-measure of 42.5%. (d) Using the PLCA model with LDS integration, reaching an F-measure of 51.4%.

IV. Evaluation

A. Training data

For constructing the pre-extracted dictionary $P(f|q,c,s)$, the DCASE 2013 Event Detection training dataset is used [6], [1]. The dataset contains isolated sounds recorded in an office environment at Queen Mary University of London, and covers 16 sound event classes ($S = 16$): alert, clearing throat, cough, door slam, drawer, keyboard click, keys, door knock, laughter, mouse click, page turn, pen drop, phone, printer, speech, and switch. Each sound class contained 20 sound exemplars. Here, the exemplar size is increased by performing data augmentation in the form of pitch shifting each isolated sound recording by ±1 semitone, resulting in $C = 60$ exemplars per sound event class, i.e. 960 exemplars in total. Using the training data, we experimented with various values for the sound state size $Q \in \{1, \ldots, 5\}$, with the best being $Q = 3$, which is used in this system. Sound state templates were extracted by providing each isolated sound ERB spectrogram as input to the NMF algorithm [17], with sparsity constraints over time in order to avoid temporal overlap of templates.

For tuning system parameters for polyphonic and monophonic sound event detection, the development datasets for the DCASE 2013 Event Detection Office Synthetic (OS)

Parameter	Values
κ, λ (sparsity parameters)	1.1, 1.4
α, β (LDS covariance factors)	0.2, 0.4
w (LDS weight parameter)	0.1

TABLE I
SYSTEM PARAMETERS ESTIMATED FROM THE DCASE 2013 OS DEVELOPMENT SET.

and Office Live (OL) tasks [6] were respectively used. Both datasets contain continuous recordings of sound events in an office environment, in the presence of background noise. The polyphonic development dataset contains 9 recordings and is synthesized by artificially concatenating recorded isolated sound events, using variable event density levels (low, mid, high) and event-to-background ratio (EBR) levels (-6dB, 0dB, and 6dB). The Office Live development dataset contains 3 continuous recordings of scripted non-overlapping sequences of office sounds, recorded at Queen Mary University of London. The system parameters estimated from the polyphonic development set are used for testing the polyphonic datasets described in the following subsection and displayed in Table I.

B. Test data

For testing, 3 polyphonic datasets of artificially concatenated office sounds were used, with varying levels of polyphony and event-to-background noise ratio (EBR). In addition, one monophonic recorded dataset of office sounds is also used, for comparative purposes.

On the *polyphonic datasets*: firstly the test dataset for the DCASE 2013 Event Detection OS challenge is used [1]. The dataset, denoted 'OS test', contains 12 recordings of 2 minutes duration each, with different event density levels and different event-to-background ratio levels. The recordings were generated using the acoustic scene synthesizer of [31] by concatenating isolated office sounds recorded at Queen Mary University of London (using different sound sources than the ones used for the OS development dataset of subsection IV-A). This polyphonic dataset allows for direct comparison with other participating systems for the DCASE 2013 polyphonic event detection task. The second polyphonic dataset uses the same event ground truth with the OS test dataset, as well as the same noise level and event density settings, but is instead generated using samples recorded at IRCCYN, École Centrale de Nantes, France. This second dataset, denoted 'OS-IRCCYN' in the remainder of the paper, is useful for evaluating the proposed method's generalization capabilities to different sound sources as well as differences in recording and acoustic conditions.

The third polyphonic dataset is also generated using samples recorded at IRCCYN, using variable event-to-background ratio levels and event density levels. The primary use of this third polyphonic dataset, denoted as 'OS-IRCCYN-2', is to test the proposed system's abilities to detect events under variable EBR and event density conditions. As for 'OS-IRCCYN', this dataset is generated using samples recorded at IRCCYN, but instead of keeping the settings of the OS dataset, several event densities and EBR are used. 3 different event density levels are used to control the event occurrences: 'low' (3 events per class), 'medium' (4 events per class) and 'high' (5 events per class). 3 levels are used to control the event-to-background noise ratio: -6dB, 0dB, and 6dB. In addition, each setting configuration (couple EBR-density) is used to simulate three scenes. For each replication, the samples to use as well as their time positions are redrawn. In order to fairly evaluate the influence of the EBR on the algorithm performances, both the samples and their time positions remain unchanged when varying the EBR. The dataset is made of 27 recordings (3 EBR x 3 densities x 3 replications), each 2 min long.

For comparative purposes, a *monophonic dataset* of office sounds is also employed, namely the Office Live (OL) test dataset from the DCASE 2013 challenge [1]. The OL dataset contains 11 scripted recordings of event sequences recorded at Queen Mary University of London, which were recorded in different acoustic environments as compared with the OL development dataset presented in subsection IV-A. Recordings in the OL test dataset have a variable duration between 1 and 3 minutes.

C. Metrics

For evaluation, we employed event detection metrics both from the DCASE 2013 challenge [1], as well as the upcoming DCASE 2016 challenge [32]. Specifically, 3 different metrics are used: frame-based (used in DCASE 2013), segment-based (used in DCASE 2016), and class-wise segment-based (used in DCASE 2016). Frame-based evaluation is performed on a 10 msec step using the post-processed event activation. For the segment-based metrics, we compare the system output and reference using a 100 msec segment size (a segment is assumed to be active if an event is detected within that segment). Finally, class-wise segment-based metrics also consider 100 msec segment size, with the results being normalized per class. A key difference between the frame-based and segment-based metrics is that frame-based metrics are computed per recording and are averaged across the entire dataset, whereas segment-based metrics count the number of true positives, false positives and false negatives across the entire dataset prior to the metrics computation [32].

In all above cases, metrics were computed using the Precision, Recall, and F-measure (P-R-F). By denoting as N_{gt}, N_{sys}, and N_{cor} the number of ground truth, estimated and correct events for a given 10msec frame, the frame-based P-R-F frame-based metrics are defined as:

$$\mathcal{P}_{\text{fb}} = \frac{N_{\text{cor}}}{N_{\text{sys}}}, \quad \mathcal{R}_{\text{fb}} = \frac{N_{\text{cor}}}{N_{\text{gt}}}, \quad \mathcal{F}_{\text{fb}} = \frac{2\mathcal{P}_{\text{fb}}\mathcal{R}_{\text{fb}}}{\mathcal{P}_{\text{fb}} + \mathcal{R}_{\text{fb}}}. \tag{15}$$

Using (15), similar metrics are defined for segment-based evaluations and class-wise segment-based evaluations. The event-based P-R-F metrics are denoted as $\mathcal{P}_{\text{sb}}, \mathcal{R}_{\text{sb}}, \mathcal{F}_{\text{sb}}$; the class-wise segment-based metrics are denoted as $\mathcal{P}_{\text{cwsb}}, \mathcal{R}_{\text{cwsb}}, \mathcal{F}_{\text{cwsb}}$, respectively.

D. System configurations - comparative approaches

The proposed system is evaluated using various configurations, namely using only the PLCA-based model of subsection III-C, using the PLCA model along with LDS-based postprocessing, and finally using the PLCA model with the LDS inference integrated in the PLCA updates, as in eq. (14). When using the LDS model, the main results presented are with the offline variant using 32 latent variables, i.e. corresponding to the random accelerations model (see Section III-D). For the PLCA-only system, the output is post-processed using a median filter with a size of 9 samples (approximately 200 msec), in order to perform smoothing over time.

In addition, comparative results are reported for the DCASE 2013 OS test set and the OS-IRCCYN test dataset using several publicly available state-of-the-art approaches for polyphonic sound event detection. All of the systems described below have been trained using the DCASE 2013 OS training and development datasets, thus results can be compared with the proposed system. The systems used for comparison are as follows:

- Gemmeke et al. [5]: based on NMF, using a frame stacking approach using time-frequency patches.
- Vuegen et al. [7]: a system using Gaussian mixture models (GMMs), with Mel-frequency cepstral coefficients (MFCCs) as input features.
- Stowell et al. [1]: the event detection baseline system from DCASE 2013, based on NMF with beta-divergence and using a constant-Q transform spectrogram as input.
- Benetos et al. [12]: a preliminary version of the proposed system, which is based on PLCA and used independent class-wise HMMs to constrain the temporal evolution of each sound state in $P(q|s,t)$. The system of [12] used a non-augmented dictionary with 20 exemplars per class. In order to make a direct comparison between the model of [12] and the proposed models, experiments are carried out with the system of [12] using the augmented dictionary presented in Section IV-A.
- A system based on the PLCA model of (4) using HMM smoothing on the sound event activation is also developed. The model, which essentially involves PLCA with HMM integration is presented in the Appendix.

Results are also reported using the DCASE 2013 OS test dataset for the polyphonic sound event detection system of Heittola et al. [3], which is based on HMM-based multiple path decoding. This system is not publicly available, thus results [3] for are those reported from the DCASE 2013 challenge for the OS test dataset.

E. Results

Sound event detection results for various configurations of the proposed system using the polyphonic OS test dataset can be seen in Table II, also compared with HMM integration within the PLCA model (presented in the Appendix). When using the frame-based F-measure, an improvement of +8.6% can be seen when comparing the PLCA-only system versus the PLCA system with LDS integration. An improvement of +7.5% is also reported when using LDS-based postprocessing over the PLCA-only system with median filtering. In terms of segment-based F-measure, the improvement over the PLCA-only system is +11.5% and +9.9% for the LDS integration and LDS postprocessing, respectively. When considering the class-wise segment-based F-measure, the improvement is +3.4% and +4.6%, respectively. The PLCA model with HMM integration outperforms the PLCA-only model, but is also outperformed by the models with LDS postprocessing and integration across all metrics.

System configuration	$\mathcal{F}_{\text{fb}}$	$\mathcal{F}_{\text{sb}}$	$\mathcal{F}_{\text{cwsb}}$
PLCA model	27.1%	30.2%	28.3%
PLCA model + HMM integration	29.9%	31.2%	29.5%
PLCA model + LDS postprocessing	34.6%	40.1%	**32.9%**
PLCA model + LDS integration	**35.7%**	**41.7%**	31.7%

TABLE II
SOUND EVENT DETECTION RESULTS FOR THE POLYPHONIC DCASE 2013 OS TEST DATASET USING VARIOUS SYSTEM CONFIGURATIONS.

System configuration	$\mathcal{F}_{\text{fb}}$	$\mathcal{F}_{\text{sb}}$	$\mathcal{F}_{\text{cwsb}}$
PLCA model	15.5%	21.7%	20.6%
PLCA model + HMM integration	20.4%	29.5%	21.4%
PLCA model + LDS postprocessing	20.0%	25.5%	21.2%
PLCA model + LDS integration	**25.9%**	**32.9%**	**22.6%**

TABLE III
SOUND EVENT DETECTION RESULTS FOR THE POLYPHONIC OS-IRCCYN TEST DATASET USING VARIOUS SYSTEM CONFIGURATIONS.

Regarding precision and recall, $\mathcal{P}_{\text{fb}} = 23.2\%$ and $\mathcal{R}_{\text{fb}} = 35.2\%$ when using the PLCA-only system. This changes to $\mathcal{P}_{\text{fb}} = 28.6\%$ and $\mathcal{R}_{\text{fb}} = 51.0\%$ when using LDS integration. This indicates that the system is generally favouring recall over precision, so the system has less missed event detections as compared to false alarms, and that LDS filtering is primarily able to improve the system's recall.

As far as the dependency of model parameters $P(s|t)$, $P(c|s,t)$ and $P(q|s,t)$ to initialisation with random values is concerned, 10 runs of the PLCA model of (4) were made using the DCASE 2013 OS dataset. The frame-based F-measure when using the PLCA model has a standard deviation of $\pm 0.2\%$, which shows that random initialisation of unknown model parameters has overall a small effect.

Results using the OS-IRCCYN dataset are shown in Table III, using various system configurations. A significant drop in performance can be seen as compared to the OS test dataset results. This can be attributed to the different recording equipment and acoustic conditions used to record the isolated sound samples, as compared to the OS test dataset. Nevertheless, the proposed LDS-based integration and postprocessing steps still demonstrate a significant performance improvement when compared to the PLCA-only system: when considering $\mathcal{F}_{\text{fb}}$, the improvement when using LDS integration is +10.4%, while when using LDS postprocessing the improvement is +4.5%. When comparing Table III with Table II, it can be seen that the performance improvement when using the proposed LDS-based methods is similar across the two polyphonic datasets. Also, LDS postprocessing is outperformed by HMM integration across all metrics for the the OS-IRCCYN dataset, although HMM integration is outperformed by LDS integration.

Table IV provides a comparison between the proposed system (using PLCA with LDS integration) and several state-of-the-art approaches for polyphonic sound event detection,

System / dataset	OS test	OS-IRCCYN
Stowell et al. [1]	12.8%	13.8%
Vuegen et al. [7]	13.5%	3.5%
Gemmeke et al. [5]	21.3%	10.8%
Benetos et al. [12]	28.0%	16.2%
Proposed method	**35.7%**	**25.9%**

TABLE IV
SOUND EVENT DETECTION RESULTS (IN TERMS OF $\mathcal{F}_{\text{FB}}$) USING THE POLYPHONIC OS TEST AND OS-IRCCYN DATASETS COMPARING STATE-OF-THE-ART APPROACHES WITH THE PROPOSED SYSTEM (USING LDS INTEGRATION).

which were described in subsection IV-D. All approaches (including the proposed system) have been trained using the DCASE 2013 OS train and development datasets and optimised using $\mathcal{F}_{\text{fb}}$, so the results can be deemed comparable. From Table IV it can be seen that the proposed system clearly outperforms other approaches, when considering both the OS test dataset (for which the sound events were recorded using the same equipment as with the OS train and development sets) and the OS-IRCCYN dataset (which is more challenging, since the recording and acoustic conditions were different when compared to the training/development sets). It is also worth pointing out that all systems (with the exception of the NMF baseline of [1]) exhibit a performance drop of -10% in terms of $\mathcal{F}_{\text{fb}}$ when comparing the OS test dataset versus the OS-IRCCYN dataset.

Results with respect to the performance of the proposed system on detecting various types of office sounds present in the OS test dataset can be seen in Fig. 6. Class-specific results vary according to the evaluation configuration: this is mostly attributed to the way the LDS observation model $\mathbf{B}$ affects the mapping of observed sound events from the PLCA model to latent sound events. This in turn affects the presence or absence of specific sound event classes in the detection output. From Fig. 6 it is seen that the proposed LDS-based postprocessing approach has the highest scores for the largest number of event classes, followed by the LDS integration. This is also explained by the class-averaged $\mathcal{F}_{\text{cwsb}}$, which is slightly higher for the LDS postprocessing method as compared to the LDS integration method.

Regarding sound classes that exhibit significant changes in terms of class-specific F-measure, the 'printer' class has an improvement of approximately +45% when comparing the LDS postprocessing method with the PLCA-only system. For the PLCA-only system, the 'printer' class has high precision (63.4%) but low recall (20.2%); the observation matrix $\mathbf{B}$ assigns several detected event classes (e.g. from 'door slam') back to the 'printer' class, which leads to a 'printer' precision of 58.8% and a recall of 92.1% for the LDS-based postprocessing approach. On the other hand, the 'speech' class drops at about -15% in terms of F-measure when comparing the two aforementioned approaches, this time because the LDS observation model $\mathbf{B}$ redistributes certain correctly detected occurrences of the 'speech' class to the 'clearing throat' class. It should be noted that the OS test and OS-IRCCYN datasets do not contain instances of the 'page turn' class (this class is however used in training the proposed system), hence it is not included in the class-wise results of Fig. 6, but is used in the computation of the frame-based and segment-based metrics.

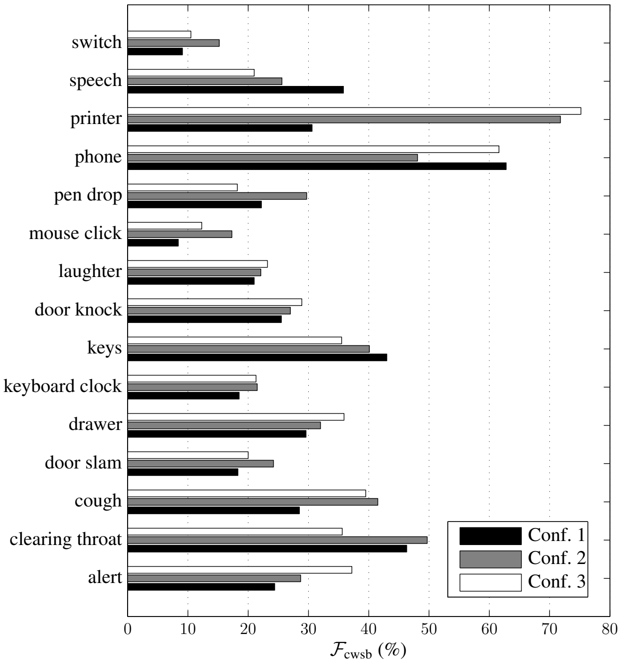


Fig. 6. Sound event detection results per sound class (in terms of $\mathcal{F}_{\text{cwsb}}$) for the polyphonic OS test dataset, using various system configurations (Conf. 1: PLCA; Conf. 2: PLCA + LDS postprocessing; Conf. 3: PLCA + LDS integration).

In order to evaluate the performance of the event tracking method under different LDS configurations, a comparison is made between the online and offline versions of LDS inference, corresponding to the Kalman filter and Kalman smoother methods (see Section III-D). When considering the online LDS with the OS test dataset, $\mathcal{F}_{\text{fb}} = 34.0\%$, and with the OS-IRCCYN dataset $\mathcal{F}_{\text{fb}} = 25.2\%$. Results when using offline LDS are at $\mathcal{F}_{\text{fb}} = 35.7\%$ and $\mathcal{F}_{\text{fb}} = 25.9\%$, respectively. This shows that performance when using the online version, which only uses information from past samples, is slightly lower as compared to using the offline method, which takes information from both past and future samples.

In order to evaluate the effect of only keeping non-negative values in the LDS posterior as part of eq. (14), a comparative experiment was carried out using the DCASE 2013 OS dataset and PLCA-LDS integration. Discarding negative values in the LDS posterior leads to a frame-based F-measure of 35.7%. However, if negative values of the LDS posterior are kept during the estimation of $P(s|t)$, $\mathcal{F}_{\text{fb}}$ drops to 33.8%.

Another evaluation of the proposed LDS-based event tracking method is made when comparing LDS with 32 latent variables, corresponding to the *random accelerations model* (see Section III-D), with LDS consisting of 16 latent variables, so only containing one latent variable per sound class. When considering the OS test dataset, $\mathcal{F}_{\text{fb}} = 32.8\%$ for the 16-variable LDS, as opposed to $\mathcal{F}_{\text{fb}} = 35.7\%$ for the 32-variable one. For the OS-IRCCYN dataset, $\mathcal{F}_{\text{fb}} = 24.7\%$ for the 16-variable LDS, while $\mathcal{F}_{\text{fb}} = 25.9\%$ for the 32-variable one. This shows that the random accelerations model is able to

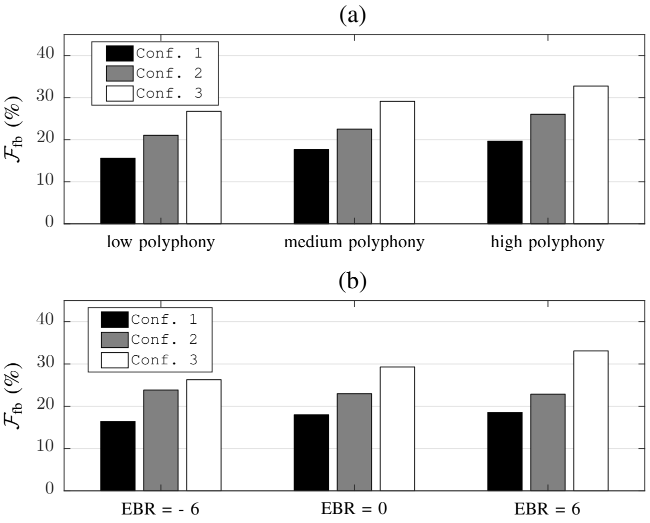


Fig. 7. Event detection results for the proposed system (in terms of $\mathcal{F}_{\text{fb}}$) on the OS-IRCCYN-2 dataset, under various (a) polyphony and (b) event-to-background (EBR) ratio levels. System configurations include Conf. 1 (PLCA), Conf. 2 (PLCA + LDS postprocessing), and Conf. 3 (PLCA + LDS integration).

provide a small but consistent performance improvement over that which considers only as latent variables the individual event activations per sound class.

Results for the polyphonic OS-IRCCYN-2 dataset are presented in Fig. 7, for groups of recordings with varying EBR noise ratio and varying event density (polyphony) levels. On average, the performance of the PLCA-only system on the OS-IRCCYN-2 dataset reaches $\mathcal{F}_{\text{fb}} = 17.6\%$; the performance of the PLCA system with LDS postprocessing is at $\mathcal{F}_{\text{fb}} = 23.2\%$; and the performance of the PLCA system with LDS integration is at $\mathcal{F}_{\text{fb}} = 29.5\%$. It can be seen from Fig. 7 (a) that for all system configurations the proposed system exhibits improved results with increased event density. In addition, the PLCA-only model and the PLCA model with LDS postprocessing are fairly stable with respect to varying EBR levels. The PLCA model with LDS integration, while outperforming the other 2 system configurations, exhibits improved results with high EBR values, i.e. with less background noise levels.

A final comparative experiment is carried out with respect to the ability of the proposed method to perform sound event detection in a monophonic scenario, using the DCASE 2013 Office Live (OL) dataset. Even though the proposed LDS-based method is mostly suited in the case of detecting overlapping events, the PLCA system with LDS integration reaches $\mathcal{F}_{\text{fb}} = 36.2\%$ as compared with $\mathcal{F}_{\text{fb}} = 32.0\%$ for the PLCA-only system. The PLCA system with LDS-based postprocessing reaches $\mathcal{F}_{\text{fb}} = 35.7\%$. This indicates that the proposed event tracking method is also useful in tracking non-overlapping events over time.

V. Conclusions

In this work, a system for polyphonic sound event detection and tracking was proposed, which combines a dictionary-based spectrogram factorisation model with a linear dynamical system. The model, which is based on probabilistic latent component analysis (PLCA), assumes that a sound event is produced as a linear combination of sound exemplars for a specific class, with each exemplar in turn consisting of a collection of ‘sound state’ spectral templates. By using the event activation output of the spectrogram factorisation model as input to a linear dynamical system (LDS) trained from fully observed data, it is possible to jointly track multiple concurrent sound events over time. In addition, by integrating the LDS-based sound event tracking process within the spectrogram factorisation-based event detection steps, it is possible to guide the convergence of the frame-based event detection model towards temporally smooth solutions. Experiments on polyphonic datasets of office sounds under variable recording conditions, event density levels, and noise/background conditions showed that the integration of LDS-based sound event tracking can lead to a substantial performance improvement over a temporally-smoothed output of a spectrogram factorisation model. At the same time, the proposed polyphonic system is able to outperform several state-of-the-art polyphonic sound event detection approaches trained on the same data. The source code[3] and created datasets[4] for the proposed system can be found online.

In the future, we will address the problem of adaptation to different acoustic conditions and sound sources; as shown by the comparison between the OS test and OS-IRCCYN datasets, a significant performance drop is reported across several event detection methods when the train and test datasets are disjoint in terms of acoustic and recording conditions. Further work on integrating supervised dynamical systems within spectrogram factorisation (and more broadly matrix decomposition) approaches will also be carried out. To that end, we will also investigate the application of non-linear and non-Gaussian dynamical systems (such as the Extended and Unscented Kalman filters [22], [15]) for the problem of polyphonic sound event tracking.

Appendix
HMM-based Model

For comparative purposes, a model for sound event detection is also developed which is based on HMMs for event tracking. The model, which can be viewed as an extension of the non-negative HMM model of [33] and is based on the PLCA model of (4), assumes that s_t is the latent state at time t corresponding to event class s, which generates observations f_t (corresponding to the observed spectra at time t). The HMM is defined by the sound event transitions $P(s_{t+1}|s_t)$, the initial sound event probabilities $P(s_1)$, and the observation model which is given by:

$$P_t(f_t|s_t) = \sum_{q_t, c_t} P(f_t|q_t, c_t, s_t) P_t(c_t|s_t) P_t(q_t|s_t), \qquad (16)$$

where $P_t(\cdot)$ denotes a time-varying distribution. $P_t(f_t|s_t)$ is essentially the approximated spectrum at time t given sound event class s_t.

[3] https://code.soundsoftware.ac.uk/projects/polyphonic-sound-event-tracking-using-linear-dynamical-systems

[4] https://archive.org/details/OS-IRCCYN

The HMM is integrated to the PLCA model in a similar way to Section III-E, where inference involves iteratively computing the PLCA posterior of (5), the sound state activation $P_t(q_t|s_t)$ from (8), the exemplar activation $P_t(c_t|s_t)$ from (10), and the sound event activation from:

$$P(s|t) \propto (w{-}1)\cdot\left(\sum_{q,c,f} P(q,c,s|f,t)V_{f,t}\right)^{\kappa} + w\cdot\gamma_t(s_t) \quad (17)$$

where $\gamma_t(s_t) = P_t(s_t|\bar{\mathbf{f}})$ is the HMM posterior, and $\bar{\mathbf{f}}$ corresponds to the complete sequence of observations. The posterior $\gamma_t(s_t)$ is computed at each iteration using the forward-backward algorithm, following the process described in [33, Ch. 2.4].

Sound event transitions $P(s_{t+1}|s_t)$ are computed in a training stage using sound event annotations from the DCASE 2013 OS Development dataset [1]. An important difference with the proposed system is that in order to learn sound event transition probabilities, the polyphonic event annotations are first converted into monophonic ones by assuming that each event is active from its onset until the onset of the next event appearing in the annotations. Initial state distributions $P(s_1)$ are assumed to be uniform. Following training with the DCASE 2013 OS Development set, weight w was set to 0.07.

References

[1] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, Oct. 2015.

[2] A. J. Eronen, V. T. Peltonen, J. T. Tuomi, A. P. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho, and J. Huopaniemi, "Audio-based context recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 321–329, 2006.

[3] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, "Context-dependent sound event detection," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2013, 2013. [Online]. Available: http://dx.doi.org/10.1186/1687-4722-2013-1

[4] J. Dennis, H. Tran, and E. Chng, "Overlapping sound event recognition using local spectrogram features and the generalised Hough transform," *Pattern Recognition Letters*, vol. 34, no. 9, pp. 1085–1093, 2013.

[5] J. F. Gemmeke, L. Vuegen, P. Karsmakers, B. Vanrumste, and H. Van hamme, "An exemplar-based NMF approach to audio event detection," in *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, Oct. 2013.

[6] D. Giannoulis, E. Benetos, D. Stowell, M. Rossignol, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: An IEEE AASP challenge," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, USA, October 2013.

[7] L. Vuegen, B. Van Den Broeck, P. Karsmakers, J. F. Gemmeke, B. Vanrumste, and H. Van hamme, "An MFCC-GMM approach for event detection and classification," *IEEE AASP DCASE Challenge*, 2013, http://c4dm.eecs.qmul.ac.uk/sceneseventschallenge/abstracts/OS/VVK.pdf.

[8] A. Mesaros, T. Heittola, O. Dikmen, and T. Virtanen, "Sound event detection in real life recordings using coupled matrix factorization of spectral representations and class activity annotations," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015, pp. 151–155.

[9] T. Komatsu, Y. Senda, and R. Kondo, "Acoustic event detection based on non-negative matrix factorization with mixtures of local dictionaries and activation aggregation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016, pp. 2259–2263.

[10] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multi label deep neural networks," in *International Joint Conference on Neural Networks*, July 2015.

[11] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016, pp. 6440–6444.

[12] E. Benetos, G. Lafay, M. Lagrange, and M. D. Plumbley, "Detection of overlapping acoustic events using a temporally-constrained probabilistic model," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Shanghai, China, March 2016, pp. 6450–6454.

[13] E. Benetos, M. Lagrange, and S. Dixon, "Characterisation of acoustic scenes using a temporally-constrained shift-invariant model," in *15th International Conference on Digital Audio Effects (DAFx)*, York, UK, Sep. 2012, pp. 317–323.

[14] C. Cotton and D. Ellis, "Spectral vs. spectro-temporal features for acoustic event detection," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, USA, Oct. 2011, pp. 69–72.

[15] K. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[16] P. Smaragdis, B. Raj, and M. Shashanka, "A probabilistic latent variable model for acoustic modeling," in *Neural Information Processing Systems Workshop*, Whistler, Canada, Dec. 2006.

[17] D. D. Li and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, Oct. 1999.

[18] M. Shashanka, B. Raj, and P. Smaragdis, "Probabilistic latent variable models as nonnegative factorizations," *Computational Intelligence and Neuroscience*, 2008, Article ID 947438.

[19] P. Smaragdis and G. Mysore, "Separation by "humming": User-guided sound extraction from monophonic mixtures," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, USA, Oct. 2009, pp. 69–72.

[20] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[21] P. Smaragdis and B. Raj, "Shift-invariant probabilistic latent component analysis," Mitsubishi Electric Research Laboratories, Tech. Rep., Dec. 2007, TR2007-009. [Online]. Available: http://paris.cs.illinois.edu/pubs/plca-report.pdf

[22] C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.

[23] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[24] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[25] C. Févotte, J. L. Roux, and J. R. Hershey, "Non-negative dynamical system with application to speech and audio," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 3158–3162.

[26] N. Mohammadiha, P. Smaragdis, G. Panahandeh, and S. Doclo, "A state-space approach to dynamic nonnegative matrix factorization," *IEEE Transactions on Signal Processing*, vol. 63, no. 4, pp. 949–959, Feb 2015.

[27] B. C. J. Moore, "Frequency analysis and masking," in *Hearing – Handbook of Perception and Cognition*, 2nd ed., B. C. J. Moore, Ed. San Diego, CA: Academic Press, 1995, ch. 5, pp. 161–205.

[28] E. Vincent, N. Bertin, and R. Badeau, "Adaptive harmonic spectral decomposition for multiple pitch estimation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 528–537, Mar. 2010.

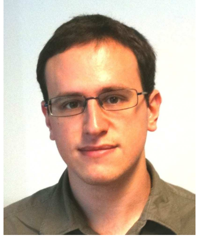
[29] G. Grindlay and D. Ellis, "Transcribing multi-instrument polyphonic music with hierarchical eigeninstruments," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1159–1169, Oct. 2011.

[30] A. N. Langville, C. D. Meyer, and R. Albright, "Initializations for the nonnegative matrix factorization," in *International Conference on Knowledge Discovery and Data Mining*, 2006.

[31] G. Lafay, M. Lagrange, M. Rossignol, E. Benetos, and A. Roebel, "A morphological model for simulating acoustic scenes and its application to sound event detection," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 2016, in Press.

[32] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, 2016. [Online]. Available: http://www.mdpi.com/2076-3417/6/6/162

[33] G. Mysore, "A non-negative framework for joint modeling of spectral structure and temporal dynamics in sound mixtures," Ph.D. dissertation, Stanford University, USA, Jun. 2010.



Emmanouil Benetos (S'09-M'12) received the B.Sc. and M.Sc. degrees in informatics from the Aristotle University of Thessaloniki, Greece, in 2005 and 2007, respectively, and the Ph.D. degree in electronic engineering from Queen Mary University of London, U.K., in 2012. From 2013 to 2015, he was University Research Fellow with the Department of Computer Science, City, University of London, London, U.K. He is currently Royal Academy of Engineering Research Fellow with the Centre for Digital Music, Queen Mary University of London, U.K. His research focuses on signal processing and machine learning for music and audio analysis, as well as applications to music information retrieval, acoustic scene analysis, and computational musicology.

Grégoire Lafay was born in 1990. He received the B.S. degree in Acoustics from the University Pierre and Marie Curie (UPMC), Paris, France, and the B.S. degree in Musicology from the Sorbonne University, Paris, France, in 2011. He received his M.S. degree in acoustics, signal processing, and musical informatics (ATIAM) from the UPMC and the Ircam in Paris, France, in 2013. Since 2013, he is a Ph.D. student at IRCCyN, Nantes, France. His research interests include acoustic scene similarity and classification as well as acoustic scene perception.

Mathieu Lagrange is a CNRS research scientist at IRCCyN, a French laboratory dedicated to cybernetics. He obtained his Ph.D. in computer science at the University of Bordeaux in 2004, and visited several institutions in Canada (University of Victoria, McGill University) and in France (Orange Labs, Télécom ParisTech, IRCAM). His research focuses on machine listening algorithms applied to the analysis of musical and environmental audio.

Mark D. Plumbley (S'88-M'90-SM'12-F'15) received the B.A.(Hons.) degree in electrical sciences and the Ph.D. degree in neural networks from University of Cambridge, Cambridge, U.K., in 1984 and 1991, respectively. From 1991 to 2001, he was a Lecturer with King's College London, London, U.K., before moving to Queen Mary University of London, London, in 2002, later becoming Director of the Centre for Digital Music. In 2015, he joined the Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford, U.K., as Professor of Signal Processing. His research interests include automatic analysis of music and other sounds, including automatic music transcription, beat tracking and acoustic scene analysis, using methods such as source separation and sparse representations. He is a Member of the IEEE Signal Processing Society Technical Committee on Signal Processing Theory and Methods.