Ego-Noise Reduction Using a Motor Data-Guided Multichannel Dictionary

Alexander Schmidt, Antoine Deleforge, Walter Kellermann

Ego-Noise Reduction Using a Motor Data-Guided Multichannel Dictionary

Alexander Schmidt[1], Antoine Deleforge[2] and Walter Kellermann[1]

***Abstract*— We address the problem of ego-noise reduction, i.e., suppressing the noise a robot causes by its own motions. Such noise degrades the recorded microphone signal massively such that the robot's auditory capabilities suffer. To suppress it, it is intuitive to use also motor data, since it provides additional information about the robot's joints and thereby the noise sources. We propose to fuse motor data to a recently proposed multichannel dictionary algorithm for ego-noise reduction. At training, a dictionary is learned that captures spatial and spectral characteristics of ego-noise. At testing, nonlinear classifiers are used to efficiently associate the current robot's motor state to relevant sets of entries in the learned dictionary. By this, computational load is reduced by one third in typical scenarios while achieving at least the same noise reduction performance. Moreover, we propose to train dictionaries on different microphone array geometries and use them for ego-noise reduction while the head to which the microphones are mounted is moving. In such scenarios, the motor guided approach results in significantly better performance values.**

I. INTRODUCTION

When a robot is moving, its rotating joints as well as the moving parts of its body cause significant noise which is referred to as ego-noise. It is a crucial problem in robot audition since it corrupts recordings and therefore degrades performance of, e.g., a speech recognizer. To relieve this problem, a suitable noise reduction mechanism is required. This task is particularly challenging because the noise involved is often louder than the signals of interest. Moreover, it is highly non-stationary as the robot performs different movements with varying speeds and accelerations. Furthermore, ego-noise cannot be modeled as a single static point interferer as the joints are located all over the body of the robot.

All this discourages the use of traditional statistical noise reduction techniques such as (multichannel) Wiener filtering [1], [2] or beamforming [3]. On the bright side however, two opportunities may be exploited. First, ego-noise is strongly structured both spatially and spectrally (c.f. Fig. 1, top) because it is produced by an automated system restricted to a limited number of degrees of freedom. Second, important extra information may be exploited in addition to the audio signals: the instantaneous *motor state* of the robot, e.g., the joints' angles and angular velocities collected by proprioceptors (c.f. for example Fig. 1, bottom).

The existence of a strong spectral structure in ego-noise motivated the use of *non-negative matrix factorization* (NMF) methods, e.g., [4], [5]. In the suppression step, the (non-negative) magnitude of a noisy speech signal in the time-frequency domain is separated into two parts. The noisy part is represented by a combination of entries from a non-negative *codebook* or *dictionary* trained on noise alone. The residual part is then used as the de-noised speech estimate. Although NMF was initially used to model the magnitude spectra of single-channel signals, multichannel extensions have been recently proposed [6], [7].

Besides methods purely based on the structure of audio signals, another approach stipulates to exploit available motor data. In [8], the time-varying noise power spectral density (PSD) is estimated by a deep neural network (DNN). The DNN is fed by motor data, which incorporates not only current, but also past sensor values. The authors show that even using moderately large networks, an increase in speech recognition rate of up to $15\,\%$ can be achieved. In [9], PSD noise templates are used for spectral subtraction. In the learning step in which ego-noise alone is present, each point in the motor data space is associated with a certain spectral noise template. In the testing (or working) phase, the approach uses a nearest-neighbour criterion to find the best matching template, which is then subtracted from the magnitude spectrum of the recording.

In this paper, we propose to fuse the information brought by a learned structured audio model on the one hand, and instantaneous motor data on the other hand, into a single framework. We propose an extension of the multichannel dictionary-learning method in [10]. The key novelty is to replace the computationally costly search in the dictionary, which is untractable for large dictionaries (NP-hard [11]), by a classification procedure guided by current motor data. These data are fed into support vector machines (SVMs) [12] to efficiently find suitable entries in an ego-noise dictionary trained beforehand. We show that this novel approach not only reduces computational complexity, but also improves performance in complex scenarios where, e.g., the head's microphone array is moving while body movements are performed.

This article is organized as follows. Section II summarizes the multichannel dictionary approach of [10]. In Section III, we present our new approach to fuse motor data with dictionary data. The results presented in Section IV provide

The research leading to these results was partly supported by the European Unions Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 609465.

[1]Alexander Schmidt and Walter Kellermann are with Department of Electrical Engineering, Chair for Multimedia and Signal Processing, Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany {alexander.as.schmidt, walter.kellermann}@fau.de

[2]Antoine Deleforge is with the INRIA center of Rennes - Bretagne Atlantique, France antoine.deleforge@inria.fr

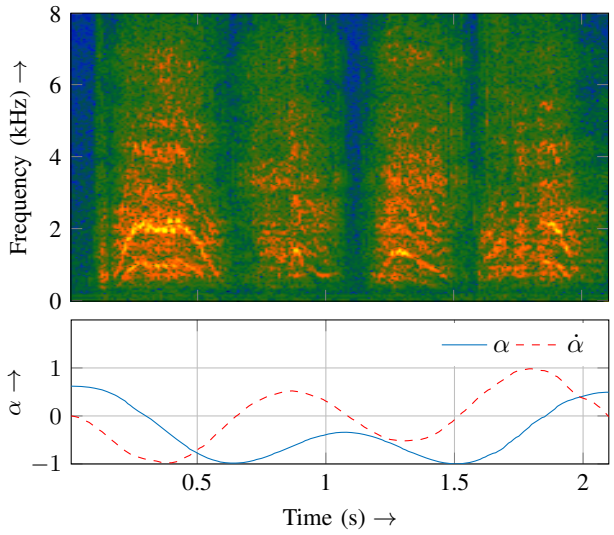

Fig. 1. Top: Spectrogram of waving right arm-movement which clearly shows spectral structured parts, Note that stationary fan noise components have been removed by a multi-channel Wiener filter as described in Section IV. Bottom: corresponding motor data, i.e., angle α and discrete-time approximations of its first derivative $\dot{\alpha}$ of involved right shoulder pitch joint (both normalized).

a proof of concept and demonstrate in which scenarios the proposed idea is advantageous and most effective.

II. Multichannel Dictionary Learning

Although ego-noise is highly non-stationary, it has distinctive spectral and spatial characteristics. The basic idea of a dictionary representation is to capture such characteristics by a collection of prototype signals, called *atoms*, collected in a dictionary. In our case, the structured ego-noise signal should be represented by a linear combination of a few atoms at each time frame. If these atoms are specifically designed to represent signals sharing spectral and spatial characteristic of ego-noise only, subtracting these atoms should remove the noise while preserving the residual signal of interest such as speech. We briefly summarize here the recent approach [10] that automatically learns a multichannel dictionary capturing both spatial and spectral characteristics of a training signal.

In the following we represent a multichannel signal in spectral domain by concatenation of the M channels per frequency bin, giving a signal vector of dimension $M \cdot F$, where F represents the number of frequency bins per channel. Then we denote the dictionary by $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_K] \in \mathbb{C}^{MF \times K}$ containing K atoms $\mathbf{d}_k \in \mathbb{C}^{MF}$, where $k = 1, \ldots, K$. Moreover, the dictionary is *corrected* by a time varying phase matrix $\mathbf{\Phi}_n \in \mathbb{C}^{F \times K}$, where each element has unit complex modulus. The *phase-corrected dictionary* is then given by

$$\mathbf{D}\{\mathbf{\Phi}_n\} := \begin{pmatrix} \mathbf{d}_{1,1} & \ldots & \mathbf{d}_{1,K} \\ \mathbf{d}_{2,1} & \ldots & \ldots \\ \vdots & \ddots & \vdots \\ \mathbf{d}_{F,1} & \ldots & \mathbf{d}_{F,K} \end{pmatrix} \odot \begin{pmatrix} \phi_{1,1,n} & \ldots & \phi_{1,K,n} \\ \phi_{2,1,n} & \ldots & \ldots \\ \vdots & \ddots & \vdots \\ \phi_{F,1,n} & \ldots & \phi_{F,K,n} \end{pmatrix},$$

where each element $\mathbf{d}_{f,k} \in \mathbb{C}^M$ captures the spectral value of atom k at frequency bin f as well as the relative phase and gain between the M channels. Here, $\odot$ denotes a modified Hadamard product, where each vector $\mathbf{d}_{f,k}$ in matrix $\mathbf{D}$ is multiplied by a global phase term $\phi_{f,k,n} \in \mathbb{C}$ in $\mathbf{\Phi}_n$.

A given multichannel spectrogram frame $\mathbf{y}_n$ should then be approximated by $\mathbf{y}_n \approx \mathbf{D}\{\mathbf{\Phi}_n\}\mathbf{x}_n$, where the vector $\mathbf{x}_n \in \mathbb{C}^K$ picks the atoms from the dictionary. Since only a few atoms should be used, $\mathbf{x}_n$ is constrained to be sparse, i.e., it should contain at most S nonzero elements, where S is referred to as the *sparsity level*. The overall problem can then be written as

$$\begin{aligned} \text{minimize} \quad & \|\mathbf{y}_n - \mathbf{D}\{\mathbf{\Phi}_n\}\mathbf{x}_n\|_2^2 \\ \text{subject to} \quad & \|\mathbf{x}_n\|_0 \leq S, \\ & |\phi_{f,k,n}| = 1, \ \forall k, \ \forall f. \end{aligned} \tag{1}$$

Here, $\|.\|_2$ and $\|.\|_0$ denote the ℓ_2- and ℓ_0-norm, respectively. The latter counts the number of nonzero elements in $\mathbf{x}_n$.

The minimization (1) is done with respect to (w.r.t.) different arguments, depending on which stage of the algorithm is considered. We briefly discuss the training and the testing stage in the following. For details, refer to [10].

Training: (1) is minimized w.r.t. $\mathbf{D}$, $\mathbf{x}_n$ and $\mathbf{\Phi}_n$. In the training stage, $\mathbf{D}$ should be learned using a set of training examples $\{\mathbf{y}_n\}_{n=1,\ldots,N_T}$. For this, [10] proposes the *phase-optimized K-SVD* (PO-KSVD) algorithm. It can be viewed as a *phase-optimized* complex extension of the popular dictionary learning method K-SVD [13]. It alternates between a sparse coding step and a dictionary update step.

Testing: (1) is minimized w.r.t. $\mathbf{x}_n$ and $\mathbf{\Phi}_n$ while the pre-trained dictionary $\mathbf{D}$ is fixed. The best fitting entries from dictionary $\mathbf{D}$ are searched and subtracted from a signal $\mathbf{y}_n$, which may contain ego-noise and speech, for example. Finding the best combination of atoms is an NP-hard problem in K and S [11], and is often approximated using iterative greedy methods [14], [15], [16]. The authors of [10] choose orthogonal matching pursuit (OMP, after [15], [16]) due to its empirically good accuracy and extends it by a computationally costly phase optimizing step. The resulting algorithm is called PO-OMP (for *phase-optimized* OMP).

III. Fusing Motor and Dictionary Data using Support Vector Machines (SVMs)

While the knowledge of the robot's instantaneous motor state should intuitively be beneficial for ego-noise reduction, an elementary question is at which stage motor data should be included. As described in previous section, one of the main bottle-necks of the multichannel dictionary method in [10] is the testing phase, where an NP-hard sparse coding problem is approximately solved by the costly iterative PO-OMP procedure. Hence, we propose to preserve the dictionary learning stage of [10], for which computational time is less of an issue, but to replace the entire testing stage by a novel and more efficient motor-guided atom selection method.

A. Fundamental idea

The physical state of a robot joint can be described by its position in terms of an angle α_n at a given timestamp t_n.

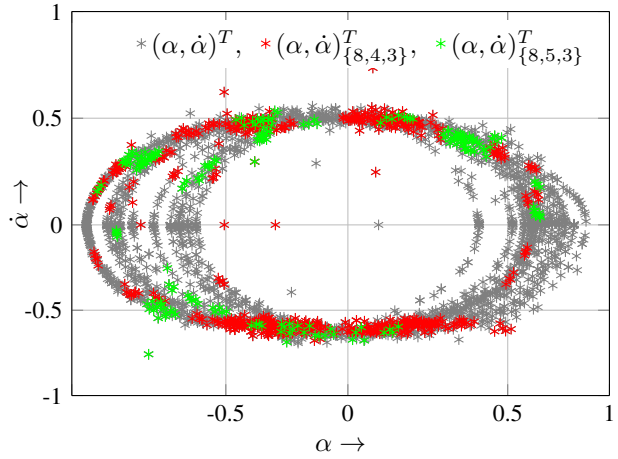

Fig. 2. Motor data points $\boldsymbol{\alpha} = (\alpha, \dot{\alpha})^T$ for right shoulder pitch movement. Highlighted points are those which are associated with a certain set of atoms, denoted as $\boldsymbol{\alpha}_{\{8,3,2\}}$ for the set of atoms $\{8, 3, 2\}$ for example. They appear to form clusters.

Furthermore, from successive angle stamps we can calculate discrete-time approximations of its first and second angle derivatives, i.e., angle speed $\dot{\alpha}_n = \frac{\alpha_n - \alpha_{n-1}}{t_n - t_{n-1}}$ and angle acceleration $\ddot{\alpha}_n = \frac{\dot{\alpha}_n - \dot{\alpha}_{n-1}}{t_n - t_{n-1}}$, respectively. For each joint, we collect the recorded and calculated angle data in a feature vector $\boldsymbol{\alpha}_n = (\alpha_n, \dot{\alpha}_n, \ddot{\alpha}_n)^T$, which is, a bit loosely, referred to as *motor data* in the following.

Assume a spectrum $\mathbf{y}_n$ is observed at a time frame in which movement noise alone is present. Additionally, the motor data $\boldsymbol{\alpha}_n$ for this frame is available. The proposed concept stipulates to associate this motor data point with those atoms PO-OMP would select in a pre-trained ego-noise dictionary to represent spectrum $\mathbf{y}_n$. Note that the order in which atoms are chosen is unimportant. For example, if at time step n PO-OMP selects atoms $8, 5$ and 7 in the dictionary and at time step m the output is $7, 8$ and 5, both motor data samples $\boldsymbol{\alpha}_n$ and $\boldsymbol{\alpha}_m$ are associated with the set of atom $\{8; 7; 5\}$. The curly brackets $\{\cdot\}$ emphasize that the order of selected atoms is not considered. The number of possible atom combinations for any given atom set is then

$$N_C = \binom{K}{S}, \tag{2}$$

If we paid attention to the ordering, N_C would be given by $K \cdot (K-1) \cdot \cdots \cdot (K-S+1)$, which is much greater. In the following, the N_C possible sets are denoted as $\hat{\mathbf{D}}_m \in \mathbb{C}^{MF \times S}$, $m = 1, \ldots, N_C$, where the choice of notation emphasizes that each $\hat{\mathbf{D}}_m$ contains a selection of atoms from $\mathbf{D}$. Our plan in the following is to decide on a set $\hat{\mathbf{D}}_m$ based on a motor data sample and use all entries of $\hat{\mathbf{D}}_m$ for ego-noise suppression.

Fig. 2 gives some motor data points in the $(\alpha, \dot{\alpha})$-plane which are associated to a certain set of atoms as an example. They appear to form clusters. The nonlinearity of the clusters' contour will motivate the use of kernel methods later on. It is reasonable to classify these points using a classifier $C(\boldsymbol{\alpha}) \in \{-1, +1\}$ deciding if a new incoming motor data point $\boldsymbol{\alpha}'$ falls into the clustering area. If yes, $C(\boldsymbol{\alpha}') = 1$, if not $C(\boldsymbol{\alpha}') = -1$ holds. All in all, N_C such classifiers must be trained. For notational convenience, they are selected in vector form

$$\mathbf{C}(\boldsymbol{\alpha}) = [C_1(\boldsymbol{\alpha}) \;\; C_2(\boldsymbol{\alpha}) \;\; \ldots \;\; C_{N_C}(\boldsymbol{\alpha})]. \tag{3}$$

The question is how such a classifier should look like. We propose to model the data points to cluster as following an unknown probability density function (pdf). By estimating its support, the above described clustering problem is solved. To do so, we refer to an idea from the wide area of support vector machines (SVM). The 1-Class-SVM method [12] estimates a classifier $C(\cdot)$ whose decision boundaries can be shown to be the support of a pdf that generated the training data with high probability. The main idea of 1-Class SVM are summarized shortly in the following.

B. 1-Class SVM

Let be given a training data set $\{\mathbf{x}_n\}_{n=1,\ldots,N_T} \in \mathcal{I}$, where N_T is the number of available training samples. In the given application these samples would be motor data points that are associated to a certain set of atoms $\hat{\mathbf{D}}_m$. This data is mapped onto a feature space $\mathcal{F}$ via a mapping function $\psi(\cdot)$ such that the training data is *separable* from the origin in the new space. *Separable* in this context means that the training data lies above a hyperplane including the origin. Note that the mapping function $\psi(\cdot)$ does not need to be known explicitly, as we will work in a kernel induced feature space. The strategy presented in [12] stipulates to find a decision function $u(\cdot)$ which is linear in $\mathcal{F}$, supports a hyperplane of the training data set and divides data from origin with maximal margin. The last property is reminiscent of the Two-Class SVM derivations in [17], [18]. More explicitly, the supporting hyperplane $u(\cdot)$ is given in its normal form by

$$u(\mathbf{x}) = \boldsymbol{\omega}^T \psi(\mathbf{x}) + b,$$

where $\boldsymbol{\omega}$ is the normal vector and b denotes the distance from the origin. T is the transpose operator. As a supporting hyperplane, $u(\mathbf{x}_n) \geq 0$ must hold for all n.

The euclidean distance from any point $\mathbf{x}$ to $u(\cdot)$ is given by

$$\frac{1}{\|\boldsymbol{\omega}\|_2}(\boldsymbol{\omega}^T \psi(\mathbf{x}) + b), \tag{4}$$

and from the origin therefore by

$$\frac{b}{\|\boldsymbol{\omega}\|_2}, \tag{5}$$

where it is assumed that $\psi(0) = 0$ is assured. Eq. 5 should be maximized or, equivalently, the reciprocal minimized.

The optimization problem is finally given by

$$\begin{aligned} \underset{\boldsymbol{\omega}, b}{\text{minimize}} \quad & \|\boldsymbol{\omega}\|_2 - b + \frac{1}{N\nu}\sum_n \xi_n \\ \text{subject to} \quad & \boldsymbol{\omega}^T \psi(\mathbf{x}_n) + b \geq -\xi_n, \; \forall n, \; \xi_n \geq 0. \end{aligned} \tag{6}$$

This follows directly from (4) and (5). As there is in general a trade-off between the width of the margin and the number

of mistakes on the training data, slack variables ξ_n are introduced. They allow misclassifications, however punish each of those wrong assignments. For this, the constraint equation is relaxed by ξ_n such that also misclassifications can fulfil it when ξ_n is chosen large enough. Moreover, ξ_n variables appear also in the optimization equation itself in which the penalisation can be adjusted by choosing ν adequately.

The optimization in Eq. (6) can be solved like any SVM problem with basic ideas from constrained optimization, i.e., formulating and solving the Lagrangian in closed form. Like in the Two-Class SVM problem, $u(\cdot)$ is defined mainly by only some samples from the whole training data set (the *support vectors*). Moreover, the final identification of the desired $\boldsymbol{\omega}$, as well as the evaluation of $u(\cdot)$ for a given data point, incorporates only inner products of the form $\psi(\mathbf{x}_n)^T\psi(\mathbf{x}_m),\ n,m = 1,\ldots,N$, which opens the door to a kernel induced feature space.

In [12], it is shown with the help of the so-called Vapnik-Chervonenkis (VC-) theory [19] that the calculated decision boundary $u(\cdot)$ is an estimator for the support of a density which generated the training data points with high probability. Using this insight, the classifier $C(\cdot)$ is defined as

$$C(\mathbf{x}) = \operatorname{sgn}(u(\mathbf{x})) = \begin{cases} +1 & \text{if } \mathbf{x} \text{ comes from the density,} \\ -1 & \text{if not} \end{cases}$$

where the $\operatorname{sgn}(\cdot)$ denotes the *signum* function, being $+1$ for positive input and -1 for negative input. It is pointed out that 1-Class SVM should exclusively be used with Gaussian kernels [17]. Its definition is very close to the one of a Gaussian density and has γ as a free parameter, the so-called kernel *width*. Among other properties, this type of kernel guarantees the separability from the origin as demanded above.

C. Ambiguity Handling

The decision regions of trained classifiers can partly overlap. Formally, an ambiguity is given if for an input motor data vector $\boldsymbol{\alpha}_n$, more than two classifiers return $+1$. A specific technique is needed to resolve such ambiguities. For this, several strategies may be considered. Here, a concept is presented that showed best performance in the experiments (see Section IV). First, all N_C classifiers are associated with a weighting factor $g_m,\ m = 1,\ldots,N_C$, being identical to the number of involved data points in each of the N_C trainings. By this, a decision region gets a higher weight when it contains more data points. Each of the K entries of the used dictionary $\mathbf{D}$ gets a counter, initialized with zero. It is then iterated over the K atoms: the counter is increased by g_m if $C_m(\boldsymbol{\alpha}_n) = 1$ and has the currently investigated atom in its recommendation. The final decision is then given by choosing those atoms that have the S largest weights.

D. Summary and Overview

This section summarizes the overall proposed methodology and provides the big picture of the training and testing stages.

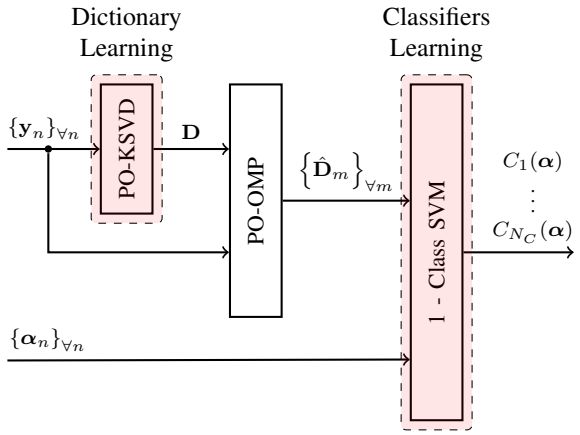

Fig. 3. Illustration of the training phase using training samples $\{\mathbf{y}_n\}_{n=1,\ldots,N_T}$ and motor data $\{\boldsymbol{\alpha}_n\}_{n=1,\ldots,N_T}$. Red shadowed areas highlight the two learning steps.

Training: The input consists in spectrogram frame samples $\mathbf{y}_n,\ n = 1,\ldots,N_T$, containing ego-noise only. Each sample is associated with a motor data vector $\boldsymbol{\alpha}_n,\ n = 1,\ldots,N_T$. After $\mathbf{D}$ is learned using PO-KSVD [10] (recall that we do not use motor data for this), PO-OMP is performed with the same samples $\mathbf{y}_n$ as input. The selected atoms per sample and the associated motor vector are then processed in the second training step which learns the 1-Class SVMs. This gives N_C classifiers, summarized in $\mathbf{C} = [C_1(\boldsymbol{\alpha}),\ldots,C_{N_C}(\boldsymbol{\alpha})]$. Each of the classifiers is associated to one specific set of atoms from $\mathbf{D}$. Fig. 3 gives a schematic overview of the training phase.

Testing: A new incoming data sample $\mathbf{y}_n$, containing both ego-noise and a target signal to denoise, and a corresponding motor sample $\boldsymbol{\alpha}_n$ are given. The latter is used to decide immediately on a set of atoms $\hat{\mathbf{D}}_m,\ m = 1,\ldots,N_C$ using the trained classifiers. The iterative search in the dictionary is unnecessary, so that the proposed algorithm can be expected to be of lower complexity than PO-OMP without motor data. What remains is only the calculation of the gains for all entries in $\hat{\mathbf{D}}_m$, collected in vector form $\hat{\mathbf{x}}_n$ and the phase optimization, resulting in the phase matrix $\hat{\boldsymbol{\Phi}}_n$. Determining those unknowns corresponds to the very last step of PO-OMP, when all atoms have been selected. Fig. 4 gives a schematic overview of the testing phase.

IV. Results

All experiments were performed on a NAO H25 robot platform, a commercial humanoid robot from *Aldebaran Robotics*, France [20]. NAO has four microphones which are all located in the head. Furthermore, the robot has 26 joints, 2 in the head, 12 in the arms, 12 in the legs. We perform exclusively movements of the right arm, including 6 joints. However, we noticed that noise of nearby shoulder and elbow joints is most prominent. A full arm movement can be subdivided into a sequence of basic motions, i.e., rising the arm, shaking, ... The complete movement consists of multiple repetitions of that sequence. Experiments have shown that good results are obtained, even if only the angle and its first

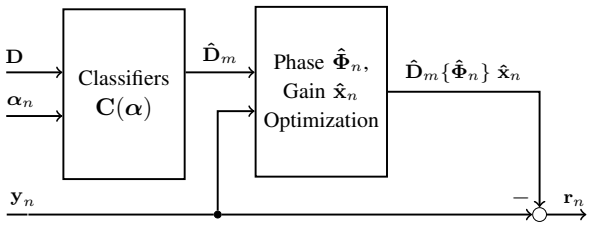

Fig. 4. Illustrated testing phase of the presented classifier approach. It selects a set of atoms $\hat{\mathbf{D}}_m$ on the basis of motor data $\boldsymbol{\alpha}_n$ only. Compared to PO-OMP this is done in a single iteration. Note that the ambiguity handling described in Section III-C is not contained in this figure.

derivative at each joint are taken. This gives a feature vector of dimension 12, $\boldsymbol{\alpha} \in \mathbb{R}^{12}$.

The sampling frequency for all recordings is $f_S = 16\,\mathrm{kHz}$, the transform to time-frequency (STFT) domain uses a Hamming window of length $64\,\mathrm{ms}$ and an overlap of $50\,\%$. NAO performed its movements in a room with moderate reverberation ($T_{60} = 200\,\mathrm{ms}$). Each dictionary used in the following was trained with $30\,\mathrm{s}$ recording. The stationary noise from a cooling fan was removed before the training started. For this, we employed a speech distortion weighted multichannel Wiener filter, as introduced in [1], [2]. It needs the power spectral density matrix of pure fan noise as input, which can be easily estimated for constant rotation speed of the fan when the robot is not moving. For testing, 200 utterances from the GRID corpus [21] were recorded with the fan switched off. The loudspeaker was positioned at $1\,\mathrm{m}$ distance of NAO, at a height of $1.5\,\mathrm{m}$. The recorded utterances were added to out-of-training movement noise. These mixtures were then used to evaluate the ego-noise suppression algorithms described above after applying the MWF to suppress the fan noise.

The classifiers were trained on 2800 motor data samples in total. To find the best parameter ν and γ, we started a sweep over different settings for both variables. Each setting was cross-checked on a set of data points which was excluded from the training.

The overall performance of the ego-noise suppression is measured in terms of Signal-to-Inference-Ratio (SIR in dB) and Signal-to-Distortion-Ratio (SDR in dB), using Matlab functions provided by [22]. While SIR measures the overall noise cancellation, SDR also incorporates information about how much speech is distorted by the suppression algorithm. Additionally, we measure keyword speech recognition rate (RR), using *pocketsphinx* [23] in the GRID corpus [21], as defined by the CHiME challenge [24].

A. Proof of concept

In a first experiment we want to show that the proposed motor guided approach reproduces results of the original PO-OMP with less computational effort. To this end, we tested different parameter constellations for K and S - the best results were obtained for a dictionary size of $K = 20$ with sparsity level $S = 3$. We parametrized the SVM with $\nu = 2^2$ and $\gamma = 2^{-2}$. Table I gives the results in terms of the presented figures of merit. Both the motor guided and PO-OMP approach clearly outperform the unprocessed recordings in all metrics used. For comparison, we also give suppression results of one-channel NMF [25]. Although NMF brings an improvement, best results are obtained using PO-OMP and the motor-guided approach. The latter clearly reproduces results of PO-OMP and slightly even outperforms it. This can be explained by the fact that PO-OMP sometimes wrongly estimates atoms due to the presence of speech (recall that PO-OMP uses audio data only). As expected, the needed calculation time in Matlab for the classifiers approach is approximately $30\,\%$ below that of PO-OMP as the search in the dictionary is unnecessary. The theoretical number of possible atom sequences and therefore classifiers is given by Eq. 2, i.e., $\binom{20}{3} = 1140$ in our case. Interestingly, in the given case only 252 classifiers must be trained as only 252 atom sets appeared. Therefore, Eq. (2) is indeed only an upper bound.

B. Variation of head position

When NAO is interacting with a human, its microphone constellation changes as it depends on the position of its head. In the following we exclusively consider horizontal head movements and denote the viewing direction with φ. For $\varphi = 0°$, NAO looks straight ahead while for $\varphi = 90°$, NAO looks to the right. With regard to training complexity it is desirable not to train dictionaries for every head position. For this, we propose a method which learns ego-noise dictionaries for a discrete number of head positions. We assume furthermore that a classifier for each dictionary has been trained. If a head position different from the trained one is tested, each of those classifiers provides a recommended set of atoms from their assigned dictionary. We concatenate those recommendations that are closest to the current head position. This results in a shortened dictionary $\mathbf{D}_{\text{short}}$ which is used for suppression.

In our experiments we used dictionaries for the discrete head positions $\varphi = 0°$, $\varphi = 30°$, $\varphi = 60°$ and $\varphi = 90°$. They are denoted by $\mathbf{D}_{\varphi=0°}$, $\mathbf{D}_{\varphi=30°}$, $\mathbf{D}_{\varphi=60°}$ and $\mathbf{D}_{\varphi=90°}$, respectively. Each of them has size $K = 20$ again. The SVMs were parametrized again with $\nu = 2^2$ and $\gamma = 2^{-2}$. The test head position was $\varphi = 45°$, therefore the concatenated dictionary is built from sets of atoms that classifiers at $\varphi = 30°$ and $\varphi = 60°$ recommend. Table II (left) gives measurement results. In comparison to Table I, it is noticeable that the baseline results, i.e., of unprocessed data, decrease further. The proposed classifier approach leads to significantly better results, especially w.r.t. speech recognition. Without knowledge of motor data PO-OMP can only use a concatenation of all trained dictionaries, i.e., $\mathbf{D}_{\text{concat}} = [\mathbf{D}_{\varphi=0°}\ \mathbf{D}_{\varphi=30°}\ \mathbf{D}_{\varphi=60°}\ \mathbf{D}_{\varphi=90°}]$. This improves results, although they stay below the motor data-guided approach.

The above described experiment can be extended if the head is moving during an arm movement which corresponds to a continuously changing microphone position. As a consequence, we cannot record the speech utterances separately and then form a mixture of ego-noise and speech. Instead, we recorded ego-noise and speech at the same time. But as the

TABLE I

PROOF OF CONCEPT

	SIR [dB]	SDR [dB]	RR[%]
Classifier	14.71	2.64	73.0
PO-OMP	14.46	2.57	71.8
NMF	2.51	0.8	45.2
Unprocessed	-5.48	-8.15	36.1

TABLE II

VARYING HEAD POSITIONS, CONCATENATED DICTIONARY, TESTING WITH HEAD LOOKING RIGHT $\varphi = 45°$ (LEFT) AND CONTINUOUSLY HORIZONTAL MOVEMENT OF THE HEAD (RIGHT)

	SIR [dB]	SDR [dB]	RR [%]	RR[%]
Classifier $\mathbf{D}_{\text{short}}$	13.6	1.78	53.0	51.1
PO-OMP $\mathbf{D}_{\text{concat}}$	12.86	1.51	39.0	40.3
Unprocessed	-6.28	-9.61	23.2	18.5

separated ego-noise and speech recordings are not available then, SDR and SIR metrics cannot be used and only speech recognition can be measured. During arm movements, NAO turns its head periodically from $\varphi = 0°$ to $\varphi = 90°$ and back again within $\approx 6\,\text{s}$. This corresponds to a rotation speed that causes only minimal ego-noise itself. The results in terms of speech recognition are given in Table II (right). The trend is the same as in experiments before: we observe a significant improvement using the motor data-guided approach.

V. CONCLUSION

We have proposed a method to fuse motor data with a multichannel dictionary approach for ego-noise suppression. In general, the complexity of the ego-noise suppression decreases while good performance values are even slightly improved. By concatenating sets of atoms which the classifier chooses, an improvement of results was also obtained for microphone constellations which were not trained. For future work, we plan to extend the dictionary approach to online-learning, and expect that the motor data-guided method will be beneficial as well. Beside this, we plan to employ bigger, commercial speech recognition engines for evaluation, as well as comparing our method to different ego-noise suppression approaches.

REFERENCES

[1] S. Doclo, A. Spriet, J. Wouters, and M. Moonen, "Frequency-domain criterion for the speech distortion weighted multichannel wiener filter for robust noise reduction," *Speech Communication*, vol. 49, no. 7, pp. 636–656, 2007.

[2] A. Spriet, M. Moonen, and J. Wouters, "Spatially pre-processed speech distortion weighted multi-channel wiener filtering for noise reduction," *Signal Processing*, vol. 84, no. 12, pp. 2367–2387, 2004.

[3] B. Van Veen and K. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE ASSP Mag.*, vol. 5, no. 2, pp. 4–24, 1988.

[4] M. Kim and P. Smaragdis, "Mixtures of local dictionaries for unsupervised speech enhancement," *IEEE Processing Lett.*, vol. 22, no. 3, pp. 293–297, 2015.

[5] M. Schmidt, J. Larsen, and F. Hsiao, "Wind noise reduction using non-negative sparse coding," in *Proc. of IEEE Workshop on Machine Learning for Signal Process.*, 2007, pp. 431–436.

[6] A. Ozerov and C. Fevotte, "Multichannel nonnegative matrix factorization in convolutive mixtures with application to blind audio source separation," in *Proc. of IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, 2009, pp. 3137–3140.

[7] H. Sawada, H. Kameoka, S. Araki, and N. Ueda, "Multichannel extensions of non-negative matrix factorization with complex-valued data," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 21, no. 5, pp. 971–982, 2013.

[8] A. Ito, T. Kanayama, M. Suzuki, and S. Makino, "Internal noise suppression for speech recognition by small robots," in *Proc. of European Conf. on Speech Communication and Technology (INTERSPEECH - Eurospeech)*, 2005, pp. 2685–2688.

[9] G. Ince, K. Nakadai, T. Rodemann, Y. Hasegawa, H. Tsujino, and J. Imura, "Ego noise suppression of a robot using template subtraction," in *Proc. of IEEE Int. Conf. on Intelligent Robots and Systems*, 2009, pp. 199–204.

[10] A. Deleforge and W. Kellermann, "Phase-optimized K-SVD for signal extraction from underdetermined multichannel sparse mixtures," in *Proc. of IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, 2015, pp. 355–359.

[11] G. Davis, S. Mallat, and M. Avellaneda, "Adaptive greedy approximations," *Constructive approximation*, vol. 13, no. 1, pp. 57–98, 1997.

[12] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[13] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[14] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Inform. Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

[15] T. Cai and L. Wang, "Orthogonal matching pursuit for sparse signal recovery with noise," *IEEE Trans. Inform. Theory*, vol. 57, no. 7, pp. 4680–4688, 2011.

[16] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Record of the 27th Asilomar Conf. on Signals, Systems and Computers*, 1993, pp. 40–44.

[17] C. Bishop, *Pattern Recognition and Machine Learning*, 1st ed. Berlin, Heidelberg: Springer, 2006.

[18] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, 1st ed. Cambridge: Cambridge University Press, 2003.

[19] V. Vapnik, *The Nature of Statistical Learning Theory*, 2nd ed. Berlin, Heidelberg: Springer Science and Business Media, 2013.

[20] Aldebaran Robotics, Soft Bank Group, "Who is NAO," https://www.aldebaran.com/en/humanoid-robot/nao-robot, 2015, accessed 7. September 2015.

[21] M. Cooke, J. Barker, S. Cunningham, and X. Shao, "An audio-visual corpus for speech perception and automatic speech recognition," *The Journal of the Acoustical Society of America*, vol. 120, no. 5, pp. 2421–2424, 2006.

[22] C. Févotte, R. Gribonval, and E. Vincent, "BSS eval toolbox user guide," IRISA, Rennes, France, Technical Report 1706, April 2005, software available at http://www.irisa.fr/metiss/bsseval/.

[23] D. Huggins-Daines, M. Kumar, A. Chan, A. Black, M. Ravishankar, and A. Rudnicky, "Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices," in *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 1. IEEE, 2006, pp. 185–188.

[24] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, "The second 'chime' speech separation and recognition challenge: An overview of challenge systems and outcomes," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2013, pp. 162–167.

[25] Y. Li and A. Ngom, "Versatile sparse matrix factorization and its applications in high-dimensional biological data analysis," in *Pattern Recognition in Bioinformatics*. Springer, 2013, pp. 91–101.