SAMbA: Speech enhancement with Asynchronous ad-hoc Microphone Arrays

Nicolas Furnon\textsuperscript{1}, Romain Serizel\textsuperscript{1}, Slim Essid\textsuperscript{2}, Irina Illina\textsuperscript{1}

\textsuperscript{1}Université de Lorraine, CNRS, Inria, Loria, F-54000 Nancy, France
\textsuperscript{2}LTCI, Télécom Paris, Institut Polytechnique de Paris, Palaiseau, France
\{firstname.lastname\}@loria.fr, \textsuperscript{2}slim.essid@telecom-paristech.fr

Abstract

Speech enhancement in ad-hoc microphone arrays is often hindered by the asynchronization of the devices composing the microphone array. Asynchronization comes from sampling time offset and sampling rate offset which inevitably occur when the microphones are embedded in different hardware components. In this paper, we propose a deep neural network (DNN)-based speech enhancement solution that is suited for applications in ad-hoc microphone arrays because it is distributed and copes with asynchronization. We show that asynchronization has a limited impact on the spatial filtering and mostly affects the performance of the DNNs. Instead of resynchronising the signals, which requires costly processing steps, we use an attention mechanism which makes the DNNs, thus our whole pipeline, robust to asynchronization. We also show that the attention mechanism leads to the asynchronization parameters in an unsupervised manner.

Index Terms: speech enhancement, ad-hoc microphone arrays, asynchronization

1. Introduction

Due to their increased number of microphones, their spatial coverage and their flexibility of use, ad-hoc microphone arrays offer great potential to speech enhancement. This potential however may be limited by a series of challenges. One of the main challenges is the need for a distributed strategy which does not rely on a fusion center as most of classic beamformers do. Distributed algorithms have been proposed for speech enhancement in ad-hoc microphone arrays \cite{1, 2, 3, 4} and recently, a deep neural network (DNN)-based distributed solution has also been introduced to combine the increased modelling capacity of DNNs with the flexibility of use of ad-hoc microphone arrays \cite{5}. Besides, because the microphones embedded in different devices do not share the same hardware and software implementations, they are acquired at different sampling rates (SRs), causing a sampling rate offset (SRO), and triggered at different starting times, causing a sampling time offset (STO). These phenomena cause asynchronization \cite{6, 7}. Asynchronization can have a negative impact on speech enhancement \cite{8}, especially on solutions relying on an accurate estimation of the direction-of-arrival, like the minimum variance distortionless response beamformer \cite{9, 10, 11}.

Solutions to asynchronization can be broadly classified into two categories. In the first category, specific signals are sent among the nodes. These signals can be either calibration signals \cite{8, 12} or time stamps \cite{13, 6, 7}. The second category gathers so-called blind approaches, because no signals are exchanged but the ones captured by the microphones of the nodes. Out of these observations, the SRO and STO can be estimated and compensated for based on the coherence \cite{14, 15}, the correlation \cite{16} or the cross-correlation \cite{17, 18, 19} between signals of different nodes. These solutions proved to be efficient, but they suffer from two limitations. The first one is that they require extra computing steps, which might overload the small devices of ad-hoc microphone arrays and add some latency to the processing. It is here interesting to note that some works showed that asynchronization could be tolerated in speech enhancement tasks without any attempt to resample the signals \cite{20, 21}. The second limitation is that none of these works studies the impact of asynchronization on DNNs although DNNs take a more and more important place in speech enhancement.

In this paper, we propose to study the impact of asynchronization on a speech enhancement solution for ad-hoc microphone arrays based on DNNs. We show that the impact of SR and sampling time offsets on spatial filtering is limited, but that their impact on the DNN performance is not negligible. To cope with this, instead of resampling the signals, we decide to use an attention mechanism which implicitly realigns the input signals of the DNN. This avoids an explicit search of the asynchronization parameters.

This paper is organized as follows. In Section 2 we describe the problem, the notations used throughout the paper and we introduce our speech enhancement system. The experimental setup is described in Section 3. In Section 4 we analyse the impact of STO and SRO on our speech enhancement system. In Section 5, we introduce a solution to cope with the asynchronization effects on the DNN performance in our system. Lastly, Section 6 concludes this paper.

2. Problem formulation

2.1. Notations

In the following, signals are considered in the short-time Fourier transform (STFT) domain, where time and frequency indices are dropped for the sake of conciseness. Bold lowercase letters represent vectors. Bold uppercase letters represent matrices. Regular lowercase represent scalars. We consider an ad-hoc microphone array of $K$ nodes of $M_k$ microphones each. The $m$-th microphone of the $k$-th node records a noisy mixture $y_{k,m} = s_{k,m} + n_{k,m}$ according to an additive noise model, where $s_{k,m}$ and $n_{k,m}$ are respectively the target speech and the noise components recorded by the microphone. The signals recorded by node $k$ are stacked in a vector $y_k = [y_{k,1}, \ldots, y_{k,M_k}]^T$.

The SRO of a node $k$ relatively to a reference node will be denoted by $\epsilon_k$. The STO of a node $k$ relatively to a reference node will be denoted by $\tau_k$.

2.2. Distributed speech enhancement in ad-hoc microphone arrays

In a previous work, we introduced a distributed speech enhancement system for ad-hoc microphone arrays, called Tango \cite{5},
It processes in two steps, highlighted in Figure 1. In the first step, at each node \( k \), a multichannel Wiener filter (MWF) \( w_{kk} \) is applied on the local signals \( y_k \). To do this, a single-node DNN (SNDNN) is used to predict a time-frequency (TF) mask \( m_k \) out of the reference signal \( y_{k1} \). The TF mask is used to compute the spatial covariance matrices of the speech and noise required by the spatial filter.

Filtering the mixture with this beamformer yields a so-called compressed signal \( z_k = w_{kk}^T y_k \). The compressed signals are exchanged among nodes, so node \( k \) receives \( K - 1 \) compressed signals \( z_{-k} \):

\[
z_{-k} = [z_1, z_{k+1}, z_{k+2}, ..., z_K]^T.
\]

In the second step, a global MWF \( w_k \) is applied on \( y_k \). The compressed signals \( z_{-k} \) are used for the spatial filtering operation, but they are also fed to a multi-node DNN (MNDNN) to predict the TF mask \( m_k \) required by the spatial filter. The spatial filters at both filtering steps are computed following the rank-1 generalized eigenvalue decomposition (GEVD) of the covariance matrices of the mixture and of the noise proposed by Serizel et al. [22].

We showed that this algorithm could efficiently process the spatial information conveyed by the compressed signals and outperforms an oracle voice activity detector (VAD)-based MWF [23]. We also showed that it performs comparatively well to FaSNet [24], while allowing for a trade-off between noise reduction and speech distortion, and relying on a much simpler DNN architecture [5]. For these reasons, we continue using this system for the current work.

### 3. Experimental setup

#### 3.1. Signal setup and DNN settings

All the signals are sampled at 16 kHz and last between 5 s and 10 s. The STFT is computed with a Hann window of 32 ms with an overlap of 16 ms. The convolutional recurrent neural network (CRNN) architecture is composed of three convolutional layers followed by a recurrent layer and a fully-connected layer. The convolutional layers have 32, 64 and 64 filters, with kernel size \( 3 \times 3 \) and stride \( 1 \times 1 \). Each convolutional layer is followed by a batch normalisation and a maximum-pooling layer of kernel size \( 4 \times 4 \) so that no pooling is applied over the time axis. The recurrent layer is a 256-unit GRU. The fully-connected layer has 257 units with a sigmoid activation function. The input of the model are the magnitudes of the STFT windows of 21 consecutive frames and the ground truth labels are the corresponding frames of the ideal ratio mask. At test time, only the middle frame of the predicted window is considered to estimate the mask, so sliding windows of the input are fed to the DNN. The mask of the whole signal is predicted before being used to enhance the speech in a batch mode.

#### 3.2. Training and evaluation data

The data used to train and evaluate our systems is extracted from the DISCO dataset. Room impulse responses in shoebox-shaped rooms are simulated. The rooms have a length, width and height randomly picked in the ranges \([3;8]\) m, \([3;5]\) m and \([2;3]\) m respectively. 2 sources, one target source and one noise source, are randomly laid in the room. 4 nodes of 4 microphones each are randomly laid in the room and record the scene. The only constraint is that the sources and the microphones should not be closer than 50 cm from each other and from the walls.

In our experiments, the effects of SRO and STO are considered separately. Their joint impact is left for future work. To simulate asynchronization, in each simulated configuration, one node \( k \) among the four is chosen as the reference node. Its SR (resp. sampling start) is left unchanged, this is why we have \( \epsilon_k = 0 \) (resp. \( \tau_k = 0 \)) for this node. The node is simulated by resampling the signals of nodes \( j \neq k \) at various sampling frequencies. The STO is simulated by padding zeros at the beginning of the signals of nodes \( j \neq k \). Because of the symmetry of the SRO and STO effects, only positive SROs and STOs will be considered. For each node \( j \neq k \), the SRO \( \epsilon_j \) is randomly taken between 0 parts per million (ppm) and a maximum value \( \text{SRO}_{\text{max}} \). 6 different values of \( \text{SRO}_{\text{max}} \) are considered, leading to 6 evaluation conditions. The STO is randomly taken between 0 ms and a maximum value \( \text{STO}_{\text{max}} \). 4 different values of \( \text{STO}_{\text{max}} \) are considered, leading to 5 evaluation conditions. Because the signals of one node share the same hardware and software implementation, we assume that they are synchronized. As a consequence, asynchronization can only affect the second filtering step of Tango.

### 4. Impact of asynchronization on DNN-based speech enhancement

In this section, the system described in section 2.2 is evaluated on the data described in section 3.2. However, the MNDNNs of the second filtering step are trained with synchronized data: at train time, the compressed signals received by a given node are perfectly synchronized with the mixtures recorded by the receiving node.

The speech enhancement performance of our system under such conditions is reported in Figure 2 in terms of SIR, SAR [25] and STOI [26], where the bars represent the 95% confidence interval. It seems from Figure 2a that the SRO has a limited impact on the performance of our system, even for high values of SROs. This is probably explained by the fact that the signals are rather short, so the effect of the SRO on the signals alignment is limited. Thus, given a rough estimation of the SRO, resampling the signals with this estimation at large intervals is enough to cope with SROs.

The impact of STO on our system is stronger. The SIR seems robust to STO, probably because of the rank-1 decompo-
Let us define a similarity matrix $S_j$:

$$S_j = \text{softmax}(\tilde{S}_j).$$

The idea of this mechanism is that $S_j$ should contain the probability of the frames of $C_j$ being aligned with the frames of the reference channel $C_1$. These probabilities are multiplied with $\{c_1(i)\}_{i=1}^T$, the columns of $C_1$ following:

$$p_j(m) = \sum_{i=1}^T s_j(m, i)c_1(i).$$

The output matrix $P_j$ of columns $\{p_j(m)\}_{m=1}^T$ is concatenated with the input matrix $C_j$ over the frequency axis.

5.2. Integration of the temporal alignment mechanism in Tango

The previously described attention mechanism is used at the input of the CRNN. Since only the second filtering step is affected by asynchronization, they are integrated into the MNDNN only. The new architecture is represented in Figure 3. Since the input data has twice more features on the frequency axis compared to the initial MNDNN, the last maximum-pooling layer has a kernel size $8 \times 1$ to keep the size of the GRU layer the same. On each node, the first channel, corresponding to the local mixture, is taken as the reference channel.

5.3. Quantitative evaluation and analysis

Three systems are compared to evaluate our solution. The first system is the same as in section 4, where the MNDNNs are trained on synchronized data only. The second system has MNDNNs trained on asynchronized data. The third system has MNDNNs with the attention mechanism, trained on asynchronized data. In the asynchronous training set, the SRO is set to 0 ppm and the STO is randomly taken between 0 ms and 32 ms. During evaluation, the STO is randomly taken between 0 ms and a maximum value $\text{STO}_{\text{max}}$. The same values of $\text{SRO}_{\text{max}}$ as in Section 4 are considered. The results obtained with these three systems are represented in Figure 4. The first conclusion from this experiment is that training the MNDNN in matching conditions brings robustness to the system in terms of SAR and STOI. However, it does not have any significant impact on the SIR. With the attention mechanism, even if the differences are not significant, there is a noticeable improvement in terms of SIR over the system where the MNDNNs are trained in matched conditions but without attention mechanism. This experiment confirms that this attention mechanism is adapted to the misalignment problem, and that it makes our speech enhancement system robust to asynchronization. Another advantage of using such a mechanism is introduced in the next section.

![Figure 3: Illustration of the attention-based CRNN.](image-url)

![Figure 2: Impact of SRO and STO on the speech enhancement performance of our system.](image-url)
Figure 4: Speech enhancement performance of three different systems.

Figure 5: 2D view of the evaluation configuration. The STO of all nodes, relatively to the first node, it also mentioned.

5.4. Qualitative evaluation and analysis

In this section, in order to highlight another advantage of using the introduced temporal alignment mechanism, we simulate a specific evaluation room to enhance the behaviour of the attention mechanism. The room configuration is represented in Figure 5 where the STOs of all nodes, relatively to the first node, are also mentioned. We represent in Figure 6 the values of the similarity matrices $\{S_j\}_{j=1}^4$ in Equation 1 computed on the first node of this room configuration. These matrices are the weights applied by the first node on the channels at the input of the MNDNN. It can be seen that these weights seem correlated with the value of the STO. For the weights applied on the second channel for example, an upper diagonal can be clearly seen, linking the $i$-th output frames with the $(i + 5)$-th input frames. Interestingly enough, the time duration of 5 frames, equal to 80 ms,\(^2\) corresponds approximately to the STO value of the second channel. Similarly for the weights applied on the third node, the clear diagonal dynamic indicates a correlation between the $i$-th output frames with the $(i - 7)$-th input frames, corresponding to a negative delay of approximately 112 ms, which is almost the STO value of the third node relatively to the first node. The same qualitative analysis can be conducted on the similarity matrix applied on the last channel.

As a conclusion of this analysis, the attention mechanism leads to a coarse estimation of the STO between asynchronized nodes in an unsupervised manner. This information could be useful to some applications which rely on a rough alignment of the signals [15].

6. Conclusions

We addressed the issue of asynchronization in a distributed speech enhancement system based on DNNs. We showed that SRO had a limited impact on our experiments, but that the influence of STO was detrimental to the speech enhancement performance of our system. To cope with it, we introduced a temporal alignment attention mechanism that makes the DNNs of our system robust to STO. In addition, we show that the hidden values of the attention mechanism can be interpreted and that they lead to a coarse estimation of the STO at all nodes. We believe that our work introduces a novel and interesting use of attention mechanisms for speech enhancement in ad-hoc microphone arrays. It would be interesting to apply this kind of attention mechanisms on signals in the time domain rather than in the time-frequency domain, where they would probably lead to more precise results and higher performance.

7. Acknowledgements

This work was made with the support of the French National Research Agency, in the framework of the project DiSCogs “Distant speech communication with heterogeneous unconstrained microphone arrays” (ANR-17-CE23-0026-01). Experiments presented in this paper were partially carried out using the Grid5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000).
8. References


