Extending Instantaneous De-mixing Algorithms to Anechoic Mixtures

Swarnadeep Bagchi
Technological University Dublin, d18128352@mytudublin.ie

Ruairí de Fréin
Technological University Dublin, ruairi.defrein@tudublin.ie

Follow this and additional works at: https://arrow.tudublin.ie/engscheleart

Part of the Signal Processing Commons

Recommended Citation

This Conference Paper is brought to you for free and open access by the School of Electrical and Electronic Engineering at ARROW@TU Dublin. It has been accepted for inclusion in Conference papers by an authorized administrator of ARROW@TU Dublin. For more information, please contact arrow.admin@tudublin.ie, aisling.coyne@tudublin.ie, gerard.connolly@tudublin.ie, vera.kilshaw@tudublin.ie.

Funder: Science Foundation Ireland
Extending Instantaneous De-mixing Algorithms to Anechoic Mixtures

Swarnadeep Bagchi  
Technological University Dublin  
Ireland  
D18128352@mytudublin.ie

Ruairí de Fréin  
Technological University Dublin  
Ireland  
ruairi.defrein@tudublin.ie

Abstract—The AdRess algorithm separates sources that are mixed using stereo, pan-mixing in a computationally efficient way. Pan-mixing gives the sources a location by introducing a relative attenuation between the versions of the sources that appear on each channel. We extend this approach so that it can de-mix sources which experience both attenuations and delays, without significantly increasing the computational burden. Separation is achieved by AdRess by constructing a frequency-attenuation matrix. We introduce a new algorithm called Delayed-AdRess. Delayed-AdRess uses frequency-delay, time-delay and frequency-attenuation matrices to separate sources from anechoic mixtures, i.e. mixing scenarios where both attenuation and delays are attenuations experienced by sources, \( x[n] \) and \( x_2[n] \). The relative attenuations experienced by sources, \( s_1[n] \) and \( s_2[n] \), as they arrive on sensor, \( x_2[n] \), are \( \alpha_1 \) and \( \alpha_2 \). Similarly the relative delays are denoted by \( \delta_1 \) and \( \delta_2 \). The observed anechoic, two-channel mixtures are

\[
x_1[n] = s_1[n] + s_2[n], \\
x_2[n] = \alpha_1 s_1[n - \delta_1] + \alpha_2 s_2[n - \delta_2].
\]  

Keywords—Source separation, anechoic demixing, time-frequency, relative attenuation, relative delay.

I. INTRODUCTION

The AdRess algorithm imposes a low computational burden and yields good separation performance for sources that have been mixed using an instantaneous mixing model [1]. The instantaneous mixing case is appropriate for professionally recorded music where sources are pan-mixed, or relatively attenuated on different channels. The problem that we consider is, in many real-world applications, the computational performance of AdRess is desirable, but AdRess does not consider time delays. In the anechoic mixing case [2], [3], when sources are attenuated and delayed by the recording environment, the application of AdRess is not appropriate. This is particularly true when sources that experience time delays experience significant phase-wrap around; estimating their mixing parameters is even more challenging. The hypothesis considered in this paper is that the AdRess algorithm, that addresses the instantaneous de-mixing problem, can be extended to the anechoic mixing case using a delay-cancellation technique.

Many Source Separation (SS) algorithms take the first step of computing the time-frequency representation of the mixtures. In a stereo setting, a synchronized short-time Fourier Transform (STFT) yields a time-frequency representation of the input signals, which is free from many multi-channel window effects [2]. DUET [3] and DESPRIT [4] compute relative attenuation and delay estimates for the versions of each source, captured on each sensor. A family of power-weighted estimators was introduced in [5], which demonstrated that these estimators can be unified into one statistical framework. These approaches are built on initial contributions in the area of Independent Component Analysis (ICA) [6]. A common theme in the ICA literature is the exploitation of the sparsity of the source signals in some transform domain [7]. If the sources are sparse, then they are in some sense already separated, and all that is required is to determine which time-frequency bins correspond to which source, in order to separate the source signals. A limitation of the hard-masking separation algorithms is that the energy in each time-frequency bin can only be assigned to one of the separated sources. The consequence of this separation strategy is that the de-mixed sources have holes in their time-frequency representations. A recent contribution, called the Redress algorithm [8], considered how to reallocate this energy to the separated sources in a way that was consistent with the relative attenuation experienced by them by reformulating the instantaneous AdRess problem as a non-negative quadratic program. We contribute an extension of the AdRess algorithm to anechoic mixtures, which does not significantly increase computational cost of AdRess. This approach is called Delayed-AdRess (D-Adress).

This paper is organized as follows. In Section II the mixing models are defined. In Section III, the AdRess algorithm is introduced. D-AdRes is defined in Section IV. In Section V, we evaluate the performance of D-AdRes by considering the task of separating real speech signals, where up to four sources are present in the mixture. The paper finishes with concluding remarks in Section VI.

II. MIXING MODELS

We consider a discrete-time, two-source, two-channel mixture case to introduce AdRess and our contribution, D-AdRess. The observations on the first and second sensor at time index \( n \) are denoted \( x_1[n] \) and \( x_2[n] \). The relative attenuations experienced by sources, \( s_1[n] \) and \( s_2[n] \), as they arrive on sensor, \( x_2[n] \), are \( \alpha_1 \) and \( \alpha_2 \). Similarly the relative delays are denoted by \( \delta_1 \) and \( \delta_2 \). The observed anechoic, two-channel mixtures are

\[
x_1[n] = s_1[n] + s_2[n], \\
x_2[n] = \alpha_1 s_1[n - \delta_1] + \alpha_2 s_2[n - \delta_2].
\]
The mixture captured at the second microphone is $x_2[n]$. It consists of the same source signals as in $x_1$, however, the sources are scaled and delayed relative to the recordings on $x_1[n]$. A synthetic source signal, $s_1[n]$, is comprised of sinusoids with frequencies 100 Hz and 300 Hz. The second source $s_2[n]$, is comprised of sinusoids with frequencies 200 Hz and 400 Hz. The first and second sources are delayed by $\delta_1 = 6.5$ and $\delta_2 = 2.5$ samples and attenuated by $\alpha_1 = 0.2$ and $\alpha_2 = 0.8$ respectively. Our goal is to recover the sources $s_1[n]$ and $s_2[n]$ from the mixtures $x_1[n]$ and $x_2[n]$. This problem is different from the instantaneous mixing problem considered by AdRess. If $\delta_1 = \delta_2 = 0$, then Eqn. 1 reduces to the instantaneous mixing problem

$$
x_1[n] = s_1[n] + s_2[n],
$$

$$
x_2[n] = \alpha_1 s_1[n] + \alpha_2 s_2[n].
$$

(2)

The original AdRess algorithm achieves good source estimates for the instantaneous mixing problem. It is not suitable for de-mixing anechoic mixtures of the form described by Eqn. 1. We continue by introducing how AdRess de-mixes sources in instantaneous mixtures, and then, we show how to extend the AdRess to the anechoic mixing case.

III. AdRess

The fundamental idea used by the AdRess algorithm is that if one mixture is scaled relative to the other one, each source in the mixtures can be cancelled from the other mixture if the appropriate relative attenuation parameter is chosen [1]. The STFT of the signal $x_1[n]$ is $X_1[k, \tau]$, where $k$ denotes the frequency bin indices $k = 0, 1, \ldots, K$. The maximum frequency bin index is $K$. The time frame indices are $\tau = 0, 1, \ldots, T$, where $T$ is the maximum time frame index. The scaling methodology used by the AdRess algorithm finds a value $g$, so that the difference between the time frequency representations is approximately equal to 0

$$
g \leftarrow \text{find} \left( |X_1[k, \tau] - g X_2[k, \tau]| \approx 0, \right.
$$

$$
\text{and } |X_2[k, \tau] - g X_1[k, \tau]| \approx 0 \right).
$$

(3)

The function, $g \leftarrow \text{find}(\cdot)$, takes one time frequency bin from each channel as its input, and returns the relative attenuation value $g$ which gives the minimum absolute difference over the range $0 < g < 1$. The relative attenuations used are $g = \frac{1}{2}, \frac{2}{3}, \ldots, \frac{G}{2}$. A frequency-attenuation matrix results from computing the absolute scaled difference in Eqn. 3 for each value of $\tau$. Fig. 1 illustrates this matrix. $A \in \mathbb{R}^{K \times M}$, for the example introduced above. This brute-force search across the feasible range of $g$ values exposes null points on the frequency-attenuation matrix $A$. In this paper, $G = \frac{M}{\tau}$, and $G = 100$. We do not evaluate $g = 0$. As the value of $G$ is increased, our estimates of the relative attenuations that give these null points improves in accuracy. Assuming that each source is located at a different attenuation position, the contribution of each source to the mixtures is recovered by constructing signals using the frequency components which have the same null locations [1]. The attenuation parameters used in Section II are $\alpha_1 = 0.20$ and $\alpha_1 = 0.80$. We observe that source $s_1[n]$ has approximately zero entries at $g = -0.20$ and similarly source $s_2[n]$ has approximately zero entries at $g = 0.80$ in Fig. 1. From Eqn. 3, it is evident that one mixture scaled relative to the other and vice versa, gives us two parts of the resultant frequency-attenuation matrix $A$, which is illustrated in Fig. 1. The negative sign is used to simplify plotting; it is used to distinguishing between both the parts of the matrix. The relative attenuation $-0.2$ indicates an attenuation location of $+0.2$ for the left hand side difference in Fig. 1. The frequency-attenuation components computed using Eqn. 3 which have nulls for approximately the same value of $g$, are used to generate the set of time-frequency components corresponding to each of the sources, e.g. $A_1$ and $A_2$ in this case. The sampling frequency is $F_s$ Hz and $T$ is the total number of time frames. The AdRess approach is unlikely to work for anechoic mixtures. In order to mitigate this, we propose the D-AdRess which incorporates delay cancellations in the de-mixing step.

IV. D-AdRess

The AdRess algorithm separates sources by creating a frequency vs attenuation matrix (Fig. 1). The D-AdRess algorithm extends this concept to derive two additional matrices namely frequency-delay and time-delay along with the frequency-attenuation matrix to estimate the delays $\delta_1$ and $\delta_2$ and attenuations $\alpha_1$ and $\alpha_2$ present in Eqn. 1.

A. Frequency-Delay Matrix

To construct the frequency-delay matrix, which is illustrated in Fig. 2 for this example, we perform the STFT with a Hamming window. In addition to the attenuation, $g$, we also test over a range of delay values $\Delta = \{\frac{1}{D}, \frac{2}{D}, \frac{3}{D}, \ldots\}$. The
delay spacing is $\frac{1}{T}$. The number of elements in delay range $\Delta$ is $N$. In the D-AdRess algorithm, we perform a brute force search over the entire delay range. For each value $\delta_i \in \Delta$, we construct a frequency-attenuation matrix $A \in \mathbb{R}^{K \times M}$, but instead of using a scalar attenuation $g$ we use $ge^{-j\omega \delta_i}$, where $0 \leq |g| \leq 1$, $\omega = -j \frac{2\pi}{N} n$, $K$ is total number of discrete Fourier transform (DFT) points and $n$ are the indices $n = 0, 1, 2, \ldots, K$.

To simplify the detection of the correct attenuation and delay for each source, we perform the follow operation on the $A$ matrix.

$$\hat{A}[i, j] = \begin{cases} \min \{A[i, :]\}, & \text{if } A[i, j] = \min \{A[i, :]\} \\ 0, & \text{otherwise}, \end{cases}$$

We sum across the columns of $\hat{A}$ to obtain the vector $\hat{a} \in \mathbb{R}^{1 \times 1}$. We perform this across all the delay values in $\Delta$. We concatenate them row-wise, and the resultant matrix is frequency-delay matrix $D \in \mathbb{R}^{K \times N}$, an example of which is shown in Fig. 2. We observe nulls at frequencies 100Hz and 300Hz are at $\delta = 6.5$ samples for source $s_1[n]$. Similarly we observe nulls at frequencies 200Hz and 400Hz are at 2.5 samples delay for $s_2[n]$. The null on 400Hz at 7.5 samples delay occurs due to phase wrapping.

### B. Time-Delay Matrix

We have evaluated the frequency-delay matrix (Fig. 2) to illustrate the operation of D-AdRess. It is not necessary to compute this matrix in practice. Instead, and to cut short our computation time, we calculate the mean of weighted-power $P = |\hat{A}|^2$ of each matrix $A$. We denote this mean $\hat{a}_{ij} = \frac{1}{L} \sum |\hat{A}|^2$, where $L$ is the total number of elements in matrix $\hat{A}$. We perform this operation across all the delay values in the set of tested delays $\Delta$. We concatenate all values $\hat{a}_{ij}$ row-wise; this gives us a vector $\hat{a} \in \mathbb{R}^{1 \times N}$. Lastly, we calculate $\hat{a}$ across all the available time frames $T$, and we stack them on top of each other, such that it takes the form of a Time vs Delay matrix, we is denoted $T \in \mathbb{R}^{T \times N}$. An example of $T$ for our worked example is depicted in Fig. 3. This entire process is summarized in Algo. 1. An example of the stacking operation and the resulting matrix is given below:

$$T = \begin{bmatrix} \hat{a}_{T1} & \hat{a}_{T2} & \ldots & \hat{a}_{TN} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{a}_{21} & \hat{a}_{22} & \ldots & \hat{a}_{2N} \\ \hat{a}_{11} & \hat{a}_{12} & \ldots & \hat{a}_{1N} \end{bmatrix}$$

In Eqn. 5 above, $T$ is the total number of time frames. In Fig. 3 the delay estimate becomes more prominent when we transform $T$ using the operation

$$T[i, j] = \begin{cases} \max\{T[; j]\} - \min\{T[; j]\}, & \text{if } T[i, j] = \min\{T[; j]\} \\ 0, & \text{otherwise}. \end{cases}$$

Fig. 3 depicts the time-delay matrix. The most distinguishable white streak is at $\delta_2 \approx 2.5$. A lighter streak is present at $\delta_1 = 6.5$ samples although it is difficult to observe in Fig. 3. The noteworthy concept here is that we can cancel out delay $\delta_2 = 2.5$ experienced by source $s_2[n]$ in mixture $X_2$, by multiplying $X_2$ with $e^{+j\omega \delta_2}$, which yields $X_2 = e^{+j\omega \delta_2} \circ X_2$. We call this delay-cancellation operation neutralization. The delay experienced by source $s_2[n]$, $\delta_2$, is now cancelled out, but this operation inadvertently delays the signal $s_1[n]$ which was already delayed by $\delta_1 = 6.5$ samples, by $\delta_2 - \delta_1 = 6.5 - 2.4 = 4$ samples. We run the D-AdRess again with the new cancelled-delayed mixture $\hat{X}_2$. In the resulting time-delay matrix, we see a prominent white streak at $\delta \approx 4$ samples. This matrix is illustrated in Fig. 4. To summarize our procedure for $\delta_1$ and $\delta_2$ estimation, we first
estimate δ₂, and then cancel out this delay in X₂ and run the same process again. Now, when we observe a discernible streak, the delay value corresponding to its location, δ, to δ₂, i.e. δ + δ₂ = 4 + 2.5 = 6.5 samples and estimate our first delay δ₁. For real speech signals from the TIMIT dataset [9], it has been observed that two streaks appear at the appropriate delay positions simultaneously. In this case, there is no need to perform the cancellation operations described above. Using the estimated delays δ₁ and δ₂, we proceed directly to estimate the attenuation parameters α₁ and α₂. Bear in mind that the angular frequency is in range −π ≤ ω ≤ +π.

In our experiments, we set G = 100, however, the greater the value of G, the greater the accuracy of estimating the attenuations α₁ and α₂. Fig. 5 and Fig. 6 illustrate estimates of the attenuations α₁ = 0.20 and α₂ = 0.80 using the frequency-attenuation matrix (in Section III). Nulls (or peaks in the transformed matrix) are created at these locations, so that the estimates g = α₁ and g = α₂ have good accuracy. Null (peaks) on the frequency-attenuation matrix are constructed by using the approach in Eqn. 6.

C. ReSynthesis

Similar to the AdRes algorithm [1], we reconstruct the sources using a portion of the frequency-attenuation matrix. This portion consists of H columns. For example, in Fig. 5 and Fig. 6 we get the null locations at attenuation positions α₁ = 0.2 and α₂ = 0.8. The peaks are located approximately at column 200 or 800 on the frequency-attenuation matrix. Setting d = 200 or 800, then we take a portion of A from d − H₂ to d + H₂. In this paper we set H = 20 columns. We sum across the columns \( \sum_{i=d-H/2}^{d+H/2} A[: i] \), such that \( A \in \mathbb{R}^{K\times1} \). We repeat this for all the time frames T. The result is a set of the magnitude vectors. These magnitude vectors are combined with the corresponding mixture (Eqn. 1) phases, i.e. \( A[k, \tau] \odot \angle X_1[k, \tau] \). We create the time-frequency representation for each source and generate a discrete time-domain source by taking the Inverse-STFT(ISTFT) of each time-frequency signal.

V. RESULTS

Four randomly selected utterances, 2 male and 2 female, from the well known TIMIT database are used [9]. The speech

Algorithm 1: D-AdRes: Time-Delay matrix.

**Input:** Analysis window length \( K \); total time frames \( T \); Number of elements in \( \Delta \) is \( N \); while \( t \leq T \) do
- for all elements in delay range \( \Delta \)
  - while \( \delta \leq N \) do
    - Sweep over the \( g \) range;
    - \( A_L = |X_1 - g e^{-j\omega_\delta} \odot X_2| \);
    - \( A_R = |X_2 - g e^{-j\omega_\delta} \odot X_1| \);
- freq-azimuth matrix is \( A = [A_L, A_R] \in \mathbb{R}^{K\times M} \).
- \( A \rightarrow \hat{A} \), to expose nulls on the freq-attenuation matrix, then find mean of weighted-power \( a_{ij} = \frac{1}{T} \sum |\hat{A}|^2 \), 1 ≤ \( i \leq T \)
- For each time window \( t \), record \( a \) and stack them one above the other, the resultant matrix \( T \in \mathbb{R}^{T\times N} \).
utterances are sampled at 16kHz. For the experiments, a 1024-
point Hamming window with a 512-point overlap was used. 
Column-1 of Fig. 7 illustrates the original utterances. Initially 
they were stereo signals. They were down-mixed to mono and 
then are remixed with two, three and four attenuation positions, 
depending upon the number of sources in our mixture. Column-2 in Fig. 7, represents the estimated sources from a mixture of four source utterances. The D-AdRess algorithm is implemented in Matlab 2019b. We use four commonly used evaluation metrics, the Source-to-Distortion Ratio(SDR), Source-to-Interference Ratio(SIR), Source-to-Artifacts Ratio(SAR) which are implemented in the BSS Eval ToolBox in order to evaluate our proposed algorithm’s performance [10]. Results are given on a dB scale. Generally, an estimated signal \( \hat{s} \) can be assumed to be composed of four components as below.

\[
\hat{s}[n] = s_{\text{target}}[n] + s_{\text{inter}}[n] + s_{\text{noise}}[n] + s_{\text{artifact}}[n] 
\]  

Here, \( s_{\text{target}}[n] \) is the true source with allowed distortions, i.e. separated source of interest. The other unwanted speech utterances that can be heard in our present estimate are \( s_{\text{inter}}[n] \). Noise that is induced by the sensors is represented by \( s_{\text{noise}}[n] \). Lastly, \( s_{\text{artifact}}[n] \) is the amount of artifacts or “burbling” sound when we hear the separated .wav utterance. These are the deformations induced by the D-AdRess algorithm while separating the source signals from the mixtures. All of these metrics, SIR, SAR, SDR are measured in dB with higher values signifying better performance. Ten Monte Carlo trials were run for each mixing scenario; the two, three and four source case. The box-plots in Fig. 8 summarize the metrics SDR, SIR and SAR achieved by both D-AdRess and AdRess in these Monte Carlo trials. The Box-plot Fig. 8 depicts in general that with increase in number of sources in the mixture, the value of these metrics tends to decrease owing to the increasing likelihood of sources overlapping in the time-frequency domain.

As the number of speech sources in the mixture continues to increase from two to four, particularly, SIRs drops from an average of 10.5 dB to 4 dB. That is, with increasing sources in the mixture, the chances of other unwanted utterances being present in our de-mixed utterance of interest goes higher. In all the scenarios, AdRess is better than D-AdRess on average by 1 dB. Higher SIR implies better separation. The SARs of both AdRess and D-AdRess for a two sources mixture are around 17.3 dB. Increasing the number of constituent sources does not significantly affect this metric. It only drops by only 0.7 dB with up to four sources. The same is the case with SDRs. Both algorithms achieve a value approximately in the range 17 to 17.5 dB for 2 source mixture, that does not vary much with increase in number of sources up to four. We observe a very negligible drop in SDRs on an average of only 0.3 dB. Thus, our proposed D-AdRess algorithm approximately matches to that of the original AdRess Algorithm in terms of SDR, SIR and SAR. Though SDR is the measure of how good the separated utterances sound, it can be at times deceiving. Thus, we evaluate our separated utterances using the Signal-to-Noise Ratio (SNR). Matlab provides a useful SNR function. The SNR’s are calculated using, \( \text{SNR} = \frac{||s_{\text{target}}||}{||s_{\text{target}}-\hat{s}||^2} \), for mixtures comprising two, three and four speaker utterances. The results are listed in Table. I. For the three sources case, a mean reconstruction SNR of 16.72 dB was achieved by D-AdRess. In comparison, AdRess achieved a mean reconstruction SNR of 16.32 dB. Again, in the four sources case, the D-AdRess achieved a mean reconstruction SNR of 17.26 dB, whereas, AdRess achieved a mean reconstruction SNR of 16.94 dB. Thus, D-AdRess delivers an improvement in the SNRs for the reconstructed source estimates. We observe that as the number of sources in the mixture increase, even though the estimated signals tend to become lower in auditory quality and intelligibility, yet there are no significant decrease in the SNR values in Table. I. On an average, D-Adress
improves the SNR for female utterances by 1.76, 1.13, 0.1 and 0.02 dB. On the other hand, the original Adress on an average improves the SNR of the Male utterances by 1.68, 0.66, 0.15 dB. We can conclude that the female utterances are separated better by D-AdRess than males. Lastly, we relied upon our human ear and it turned out that for the two source mixture, the separated sources were very high both in auditory quality and intelligibility. As we increased the number of sources in the mixture to four (Fig. 7), the auditory quality and intelligibility of the separated sources diminished and the Female2 tended to contain some evidence of Male1. The rest of the separated sources were of good quality.

A. Computational Complexity

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Female1</th>
<th>Male1</th>
<th>Female2</th>
<th>Male2</th>
</tr>
</thead>
<tbody>
<tr>
<td>D-AdRess</td>
<td>14.56</td>
<td>16.87</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>AdRess</td>
<td>14.44</td>
<td>18.55</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>2-Source</th>
<th>3-Source</th>
<th>4-Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>D-AdRess</td>
<td>14.44</td>
<td>17.76</td>
<td>14.44</td>
</tr>
<tr>
<td>AdRess</td>
<td>14.31</td>
<td>18.44</td>
<td>16.23</td>
</tr>
</tbody>
</table>

Thus, our proposed D-AdRess algorithm is able to tackle the delay in arrival of a particular signal source (anechoic mixtures) that had not been tackled by the original Adress instantaneous model. In terms of SDR, SIR, SAR and SNR, the D-AdRess is approximately 90-95% matches to that of Adress. D-AdRess delivers an improvement in the average SNR for the reconstructed source estimates. Until now we have considered small delays in the range of 1 to 10 samples. We need to extended our D-AdRess to big-delay models so as to make it more apt for real-life sensor network-type substrate. We postulate that the performance gains achieved by D-AdRess will be enhanced as the delays experienced by the signals increases, as Adress was designed in order to demix sources which were mixed using an instantaneous mixing model. Also, we need to reformulate D-AdRess as a soft masking model, so as to reallocate appropriate amount of TF energy to each of the separated sources based upon their attenuation positions in the stereo field.

VI. CONCLUSION AND FUTURE WORK

This paper has emanated from research supported in part by a Grant from Science Foundation Ireland under Grant number 18/CRT/6222. This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under the Grant Number 15/SIRG/3459.

ACKNOWLEDGMENT

TABLE I

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>2-Source</th>
<th>3-Source</th>
<th>4-Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>D-AdRess</td>
<td>14.44</td>
<td>18.37</td>
<td>18.53</td>
</tr>
<tr>
<td>AdRess</td>
<td>14.42</td>
<td>18.52</td>
<td>17.40</td>
</tr>
</tbody>
</table>

Fig. 8. SDR, SIR, SAR of D-AdRess and Adress grouped by number of sources in the mixture, Column1, Column2 and Column3 corresponds to 2-Sources, 3-Sources, 4-Sources respectively due to the nested loop in Algo. 1. So, the resultant excess cost incurred by D-AdRess compared to Adress is $O(N \times M) - O(M)$ and is not that significant provided, $M$ and $N$ are not very high in value. In our project we have used $M = 100$, and $N = 10$.

REFERENCES