高度交通システム　8－12
(2002.　3.　5)

車内雑音環境下における音源分離手法の検討

Yann LY-GAGNON[†]　　高橋　真之[‡]　　前田　典彦[‡]

[†] École Polytechnique de Montréal
Montréal, Québec Canada
[‡] NTT サービスインテグレーション基盤研究所
〒239-0847　横須賀市光の丘 1-1

E-mail: [†] yann.ly-gagnon@polymtl.ca, [‡] {takahasi,maeda}@nttmhs.tnl.ntt.co.jp

あらまし 車内で利用する音声対話システムにおいて音声認識率の向上を図るため，2 本のマイクロホンと DSP を用いて音声 S/N を改善するための検討を行った．音声とガウシアンノイズによるシミュレーション及び実際の車内における収録実験により適用アルゴリズムによる雑音抑圧効果を評価した．シミュレーションでは約 16dB の SN 比改善が見られた．

キーワード ブランド音源分離，車載環境，雑音抑圧，音声認識

A Study on Blind Signal Separation for ITS

Yann LY-GAGNON[†],　Masayuki TAKAHASHI[‡],　and　Fumihiko MAEDA[‡]

[†] École Polytechnique de Montréal
Montréal, Québec Canada
[‡] NTT Service Integration Laboratories
1-1 Hikarinooka, Yokosuka, Kanagawa 239-0847 Japan

E-mail: [†] yann.ly-gagnon@polymtl.ca, [‡] {takahasi,maeda}@nttmhs.tnl.ntt.co.jp

Abstract Aiming the improvement of speech recognition rate in vehicle's environment, this paper examines blind signal separation algorithm in vehicle's environment with two microphones and DSP. We carried out two experiments applying the algorithm, simulating on the PC with voice data and Gaussian noise, and recording real voice in noisy environment of running vehicle. About 16dB of SNR improvement is observed in the theoretical simulation.

Key words blind signal separation, in vehicle's environment, noise suppression, speech recognition

1. Introduction

One of the main reasons of the slow proliferation of speech recognition applications is the ambient noise. This noise is even more present if we are using mobile communications. Usually, noise contaminates sources and degrades the quality of the speech signal making it difficult for the speech recognition application to function normally.

In this paper, we present a study, validation and implementation of the output decorrelation algorithm for blind signal separation.

The implementation of the algorithm is to provide a concrete solution for improving the speech recognition results in adverse environment. The immediate application is to provide a solution for ITS so that the driver is able to communicate with the navigation car system without being distracted by the LCD and keeping his eyes on the road.

In the next few years, on-board car computer systems will allow us to be entertained and more productive when driving. However, security might be compromise. For example, if a user wants to browse the web, a VoiceXML engine would enable him to request information. However, many external sounds sources can degrade the speech recognition such as the engine or the wind whistle. The goal of our project is to provide a preprocess unit dedicated to separate the speech from the noise, the same way the brain does while he is at a cocktail party.

2. Possible algorithm for the output decorrelation

2.1 Algorithm specifications

There are different approaches to solve this kind of problems. This step in the design process was done after establishing the general specifications. Within those specifications, we are making many simplification and generalisation about the problem. These simplifications can be a cause why our algorithm might perform in real situations.

Here are some approaches we have considered:

- Blind signal separation vs Blind deconvolution

For the mixing system (only for theoretical simulations)

- Modelisation
- Linear or non linear
- Scalar vs. Convolution

For the demixing system

- Parameter's convergency
- Number of sensors and sources

2.2 Blind signal separation vs Blind deconvolution

The definition of the blind signal separation is that we have mutually independent unknown signals. These signals are also mixed in an unknown environment. We are trying to find a demixing matrix whose solution corresponds to the original signals.

The definition of the blind deconvolution is to find the inverse system which can give the original sources. The difference from the blind signal separation is that usually it only involves a single source observable source [1].

2.3 Mixing system

Modelisation:

Another aspect which we have to be careful of is the modelisation or not of the sources. We must remember that we don't have any information about the source signals. In the literatures, the common model used is the AR Model [2]. Another possibility was to use the EM algorithm to search the parameters. Since these two techniques need a lot of effort and time, with maybe little improvement, we will not choose to model our original signals.

Linearity:

We will assume that the sources relation from the input and the output is linear and the statistical properties are constant during the processing blocks (we will talk about it later in the chapter). We do this in view to simplify our analysis.

Scalar vs convolutive:

When doing the theoretical simulation, we can combine the signals using a matrix filled with random scaling factor. This can be unrealistic in real situation where we find delays and non-linear effects can be included in the voice. This can be called scalar mixing.

2.4 Demixing system:

This is the part that represents the heart of our problem. We are trying to find the original signals by using (and making assumption) about a inverse mixing model. There are many different attempts done to realize this in the literature. We will go through some of the most frequently used and motivate our choices why we took the output decorrelation method.

Finding the parameters of the demixing model can be done by recursive or iterative algorithm. Our algorithm uses the iterative model.

Another critical issue we have to look at is if these parameters are stationary or adaptive. We will make the assumptions that the signals are stationary to simplify the computational complexity. However, this can bring very bad results if they are not stationary. This problem could be resolved by overlapping blocks.

Using adaptive parameters, this implies having a new parameter called learning rate. Also, this learning rate is usually not constant, so this increases the complexity of the algorithm and the processing.

The demixing systems can be:

- Blind output decorrelation
- Mininum Mutual Information MMI and Maximum entropy
- Linear
- Feed forward/ backward structure
- Neural networks
- Geometry based

3. Outline of the algorithm used by our function

Compare to the other algorithms, the technique of the output decorrelation has relatively moderate computational complexity enough to implement the solution to a DSP.

3.1 Ideal blind signal separation

The blind signal separation tries to exploit the spatial properties of speech and noise those are picked up by multiple sensors (microphones). Figure 1 shows the principal of the ideal blind signal separation. N independent vectors X, we try to estimate the demixing matrix W to find the initial independent vectors U. We define the output by:

$$Y=WX=W(AU) \qquad (1)$$

3.2 Cross correlation and the goal of the algorithm

Figure 2 shows the operation of the blind signal separation. In this study, we assume using two microphones in the car. The goal of this algorithm is to minimize the cross-correlation C between two mixed inputs, to find the theoretical value of y1 and y2.

The biggest reason why we chose this algorithm was for his simplicity and good performance. This algorithm is to be implement in DSP solution which needs to be performed real time computation and have a limited MIPS power processing. Also, KOUTRAS A., DERMATAS E. and KOKKINAKIS G. showed experimentally that the output decorelation method works better than the Maximation Information. [3]

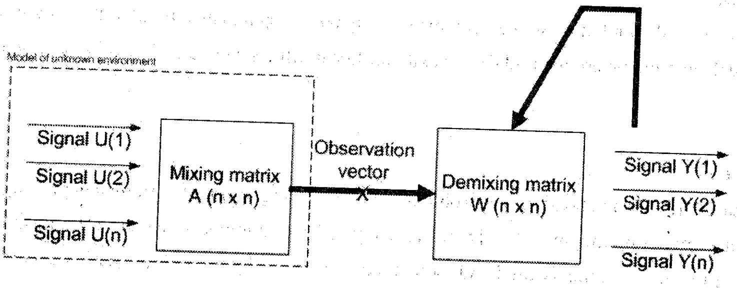

Figure 1. Principle of the ideal blind signal separation

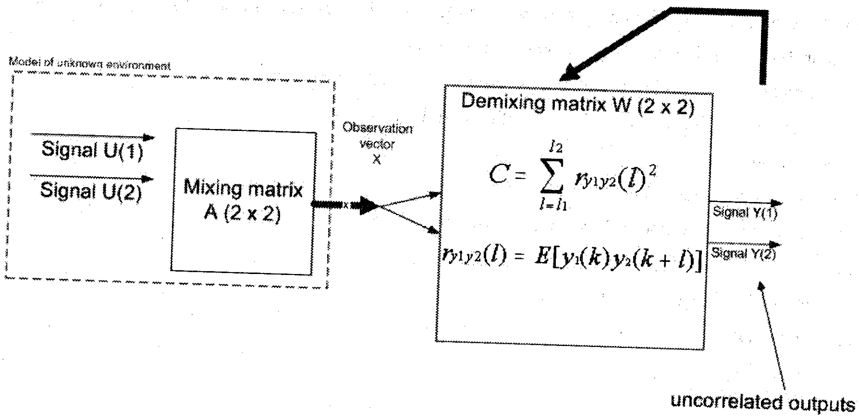
Model of unknown environment
Signal U(1)
Signal U(2)
Mixing matrix
A (2 x 2)
Observation
vector
X
Demixing matrix W (2 x 2)
$C = \sum_{l=l_1}^{l_2} r_{y_1y_2}(l)^2$
$r_{y_1y_2}(l) = E[y_1(k)y_2(k+l)]$
Signal Y(1)
Signal Y(2)
uncorrelated outputs

Figure 2. Operation of the blind signal separation

4. Implementation of the algorithm and simulations

For the simulations we are only considering the case of 2x2 separations. Even if the literatures mention the possibility to apply this to more than two inputs, our first goal is to study to feasibility of the implementation on a DSP.
During the simulations the parameters to change are the order of the tap FIR filter and the correlation lag (l). Those simulations must be executed case by case to verify the speech recognition ratio. Even with the help of scripts, this process can be very exhausting.

4.1 Theoretical simulation

The theoretical simulation is realized with Gaussian noise and a man speaking at different sampling rates (32 kHz, 16kHz and 8 kHz). We will consider only the sampling at 8 kHz that proves the robustness of our algorithm.
We are using a FIR Filter to mix the two original signals. FIR filter are very popular for this kind of situation since they are able to modelise the delays, the echoes and the reverberations. The Mixing and the demixing process using the algorithm were implemented to Matlab program running on PC.
Also we will quantify the quality of the separation by calculating SNR. The experimental results show us that we have

achieved a relatively good separation. When we listen to the sound output, there almost no difference compare to the input. This is important to mention, because as I will talk later, some distortion effects seems to appear during the simulation of the real samples.

Notice, to achieve these results, we had to change algorithm parameters manually and verify manually if the results are acceptable or not. These parameters include the filter order and the correlation lag. For the theoretical simulation these parameters were set arbitrarily at:

Filter order = 4

Correlation lag = 20

Of course if we change those parameters, the results are likely to change.

The analysis of the frequency spectrum shows us that there is almost no either for the separated voice. Notice that the Gaussian separated noise is also well distributed across the spectrum. The noise is also almost uniformly distributed.

Figure 3 shows the time domain of original voice signal and that of separated one after demixing. Figure 4 shows the frequency domain analysis of those signals.

4.2 Discussion about the result for the theoretical simulation

For the theoretical simulations, it is possible to calculate the SNR (Signal noise ratio) since we know what is the original signal. This gives us an idea about the improvement of the separation.

The SNR is (in dB):

$10 * \log_{10}(\text{Power of signal} / \text{Power of Noise})$.

Where the gain is:

$\text{GAIN} = \text{SNR}_{\text{separated signal}} - \text{SNR}_{\text{mixed signal}}$.

To calculate the noise, we are substracting the original signal from the separated signal. Also, we must be careful about having the same amplitude for inputs and outputs.

The improvement for the theoretical simulation is about 16 dB. Note that this value varies a little bit since the separation is always slightly different.

For example:

We had: $\text{SNR}_{\text{separated signal}} = 19.50$ dB

$\text{SNR}_{\text{mixed signal}} = 2.79$ dB

Gain = 19.50 dB - 2.79 dB = 16.72 dB

4.3 Real simulations

This is by far the harder to process and the more realistic situation. The car type used in these experiments is a Honda Civic Step Wagon.

The recordings were done with a SONY PCM-M1 DAT recorder with a sampling of 44.1 kHz. Even if the spectral content after 15 kHz could be consider negligible (and even less if we are using other telephony architecture), we are not doing any filtering before putting the two signals in our algorithm. Two microphones are placed on the dashboard at 30 cm to each other. They are placed near the driver head's. The car is running in 80 km/h on a highway. The microphone type used is a C 400 PC from AKG. The mixing and the demixing process were executed by Matlab same as theoretical simulations.

The separated outputs have delays and reverberations those are hard to see on graphics but easy to notice when we are listening to the sound samples. This is particularly the case for the example with distortion.

Figure 5 shows the time domain of original signal recorded in the vehicle and the output signal that is separated by the algorithm. Figure 6 shows the frequency domain analysis of those signals.

4.4 Real simulation separation example

With the results of this technique we are able to improve the SNR. Notice that it is possible to have better results, but these comes with distortions that are very bad for speech recognition.

5. Future studies

Many points are still to ameliorate about this technique for blind signal separation before any commercial applications can be created. Here's a review of these principal points.

5.1 Simulations coherence

When we do our simulations, we divide our time sample in time stamps where we are varying the different parameters. We are looking for any parameter combination to find the best results. However, we haven't found any parameter combinations that allow a maximal separation. In a future study, we will have to find a relation between the parameters and the quality of separation.

5.2 Frequency distortions

For some specifics conditions, the separation comes with so much that it comes impossible to apply in the speech reconnaissance engine.

In a future study, we will need to find the causes of these frequency distortions and find how can we overcome this problem.

5.3 Technical problems

While doing scripts automating the large amount of simulations for data, we encounter some technical problems. The most difficult problem to solve was the Out of Memory errors. We found out that Matlab doesn't really manage memory since the C compiler (malloc, calloc, free) and the operating system have control over the available RAM. This was specifically the case on the Windows NT Workstation. To save memory, we tried to declare the big matrix once (as a constant in C) to avoid fragmentation of the memory. This way, declaring the matrix assures that Matlab won't use its internal functions to allocate new memory for this matrix. We also had to break our simulations into discrete parts.

6. Conclusion

In this paper, we examines blind signal separation algorithm for noisy in-vehicle environment. with two microphones and DSP. We carried out two experiments. Simulating mixing and demixing on the PC with clear voice data and Gaussian noise revealed good result of 16dB of SNR improvement with almost no frequency domain distortion. Processing real voice data recorded in noisy environment of running vehicle shows relatively small SNR improvement with some frequency distortions. We confirmed utility of algorithm under particular conditions but further studies are needed for real commercial application.

Acknowledgements

We would like to make our acknowledgements to Honda R&D Co., Ltd for supporting in-vehicle experiments.

References

[1] BELL A.J. & SEJNOWSKI T.J. "An information maximization approach to blind separation and blind deconvolution", Neural Computation, 7, pp.1129-1159, 1995

[2] MAKHOUL, John. "Linear prediction a tutotorial review" IEEE Proc. 63(4), pp. 561-580, April 1975

[3] KOUTRAS A.,DERMATAS E. and KOKKINAKIS G., "Speech Recognition in a real room and multi-simultaneous-speaker environment", 1st International Workshop on Text, Speech & Dialogue TSD, pp. 251-256, Brno, Czech Republic, September 1998.

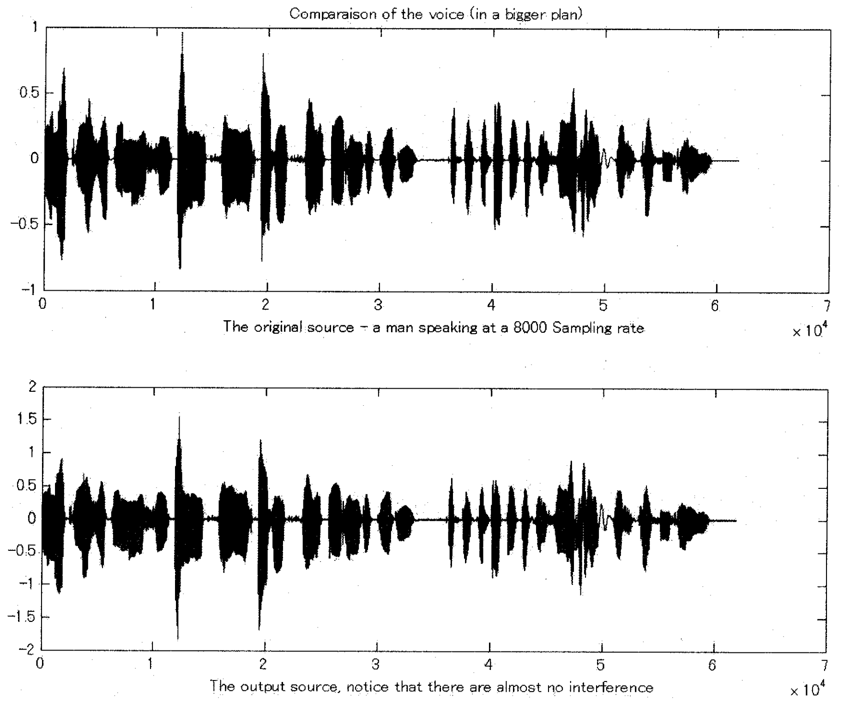

Figure 3. Original source and the separated source on theoretical simulations

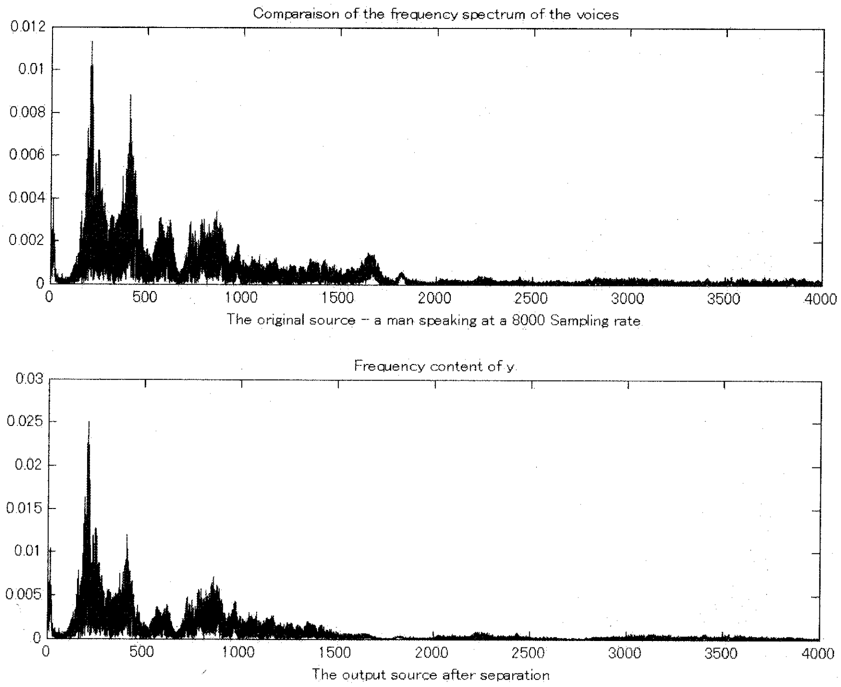

Figure 4. Frequency analysis of the original voice and the separated voice on theoretical simulations

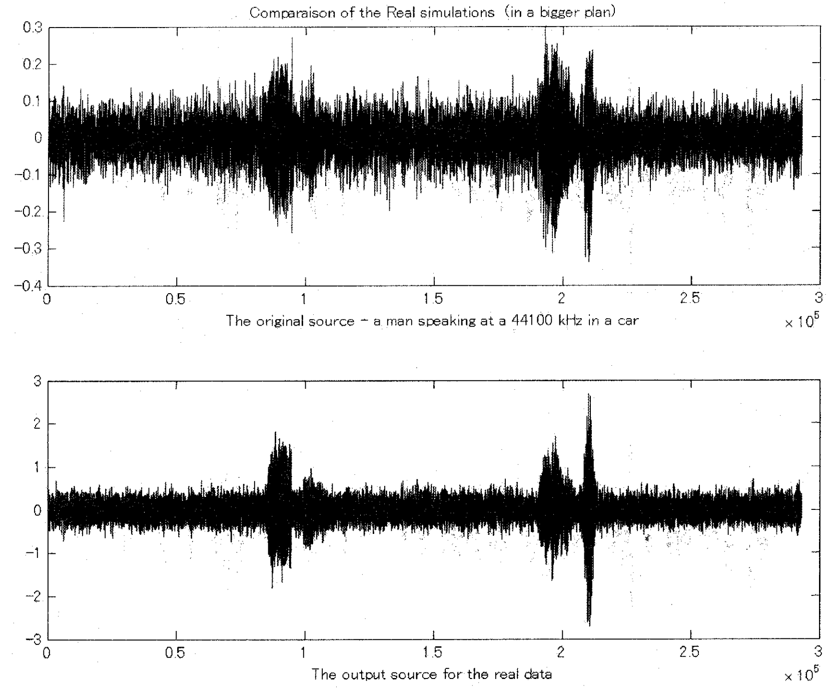

Figure 5. SNR improvement on real simulations

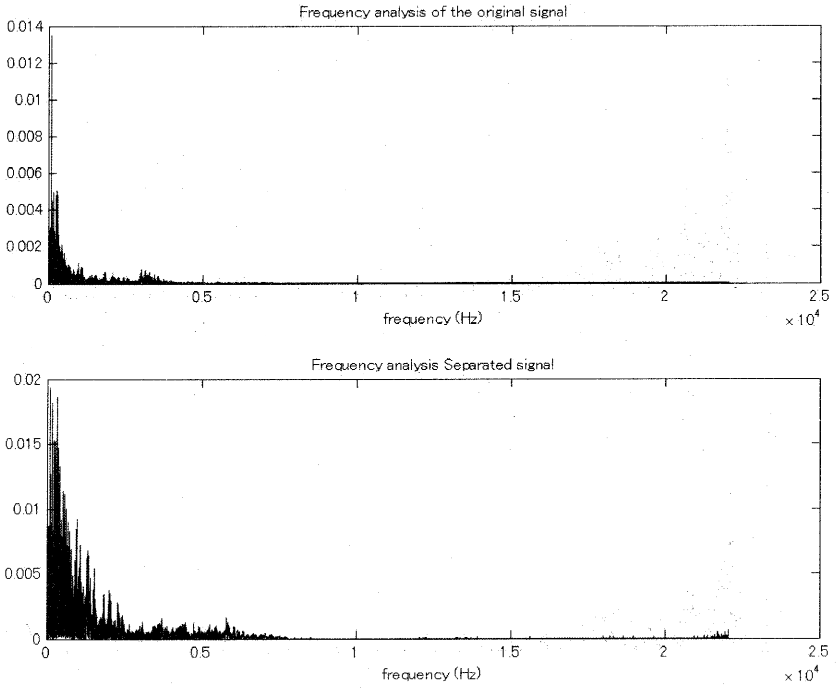

Figure 6. Frequency analysis of the separations on real simulations