EFFECTS OF FREQUENCY DISPLACEMENT IN INDEPENDENT COMPONENT ANALYSIS

Caleb Rascon
University of Manchester
School of Electronic and Electrical Engineering
Manchester, UK, M60 1QD
caleb.rascon@postgraduate.manchester.ac.uk

Barry Lennox
University of Manchester
School of Electronic and Electrical Engineering
Manchester, UK, M60 1QD
barry.lennox@manchester.ac.uk

Ognjen Marjanovic
University of Manchester
School of Electronic and Electrical Engineering
Manchester, UK, M60 1QD
ognjen.marjanovic@manchester.ac.uk

ABSTRACT

Independent Component Analysis (ICA) is widely used for Blind Source Separation in generic spectra which are themselves obtained from sensors that can be decalibrated or are too sensitive to ambience changes. This usually results in frequency displacement or lag that ICA will face during its source extraction. Experiments were done that show that ICA is not well-equipped to handle such displacement, and that it only is able to extract the same components as before being lagged given only an insignificant amount of displacement. Other experiments showed that the amount of lag that ICA can handle varies depending on the width of the components intended to be extracted.

Keywords: Independent Component Analysis, Shift, Spectral Analysis

1 INTRODUCTION

Independent Component Analysis (ICA) has been used extensively throughout different areas of the industry [2, 3]. One of its most important applications is the identification of components of a mixture produced by a plant, by analysing an energy spectrum that was measured from the mixture. This is fundamental for quality control, where ICA can aid the process of confirming the presence of important materials inside the mixture, and by identifying the presence of unwanted materials.

Unfortunately, the spectra that ICA analyses may suffer from frequency displacement, or lag. It has been shown that such phenomenon often occurs in real life applications as a result of poor sensor calibration and/or external influences [4]. The temporary solution for this of constantly re-calibrating the spectral sensor is known to be costly, as expensive materials and plant downtime are necessary [4]. The approach of modelling the external influences as a mean to counter it [5] is unrealistic, as information (such as concentration of each component, temperature at moment of sampling, etc.) needed to build such a model is rarely available. In addition, because frequency displacement may appear in small quantities, the monitor in charge may not be even aware of such occurrence. Adding all of this together, lagged measured spectra obtained from the industry is very common, so the possibility of the spectrum being shifted needs to be accounted for as part of its analysis.

It is then important to investigate what are the effects of applying ICA to spectra that are known to be lagged. And if such effects are detrimental, understanding them would be a first step towards making ICA robust against them.

In this paper, ICA will be put to the test with a series of artificial data sets, who were subjected to different amount of lags. This will be done to find out what is the range of lag that ICA can handle before obtaining a set of components different from the set it obtained from the same non-lagged spectra.

Section 2 will give a brief background of ICA, centering around the FastICA algorithm. Section 3 explains the methodologies used for creating the test data and how it was artificially lagged. Section 4 provides details about the several tests and experiments done with the FastICA algorithm. Finally, Section 5 gives some conclusions and a brief discussion about the results given in the experiments.

2 BACKGROUND ON INDEPENDENT COMPONENT ANALYSIS AND SHIFT

The objective of this section is to refresh the memory of the reader towards how ICA works, as it is essential for understanding how a lagged spectrum may affect its performance. ICA was first introduced by Comon in 1994 [1] and it is one of the principal methodologies behind Blind Source Separation (BSS), the objective of which is to obtain a set of sources from analyzing only a group of mixtures of those sources.

Its main assumption is that the sought-after sources are *independent*. This assumption is viable, as in many applications the presence of one component in a mixture is not dependent on the presence of another. Independence between two sources, in the practical sense, means that there is little or no mutual information between them. Hyvärinen in 1997 proposed a way to measure it and called it *negentropy* [11]. This measure of independence was used [12] as an objective function in a fixed-point iterative algorithm to find an appropiate de-mixing matrix, with which estimates of the sources could be extracted from the mixtures. The estimates, because of the use of negentropy, have very little mutual information between them, in effect being independent (thus, called Independent Components, or ICs). This gave birth to the FastICA algorithm, which is one of the most popular implementations of ICA and has been used extensively in different areas of science [6, 7, 13].

In a tutorial paper about ICA, Hyvärinen wrote that "Actually, and perhaps surprisingly, it turns out that [to solve the ICA problem] it is enough to assume that [the sources...], *at each time instant* t, are statistically independent."[13] This implies that the sources need to be 'aligned' throughout the different mixtures for such independence to be measured accurately. If this alignment is not present, incorrect measures of independence will arise, and one source that is lagged in one mixture and not in another may be considered as two independent components instead of one.

3 METHODOLOGY

FastICA was applied to a series of artificial data sets created from a set of simulated spectral components, acting as a set of reference spectra. For ease of visualisation, each spectral component in the reference set was made up of just one 'peak'. Such peaks were created using (1).

$$c_i = \frac{1}{0.2h\sqrt{2\pi}} e^{\frac{-(f_i-p)^2}{2w^2}} \quad (1)$$

where c is one component; c_i is the energy at frequency f_i; h, p, and w are the height, frequency location, and width of the peak respectively. These peaks could then be artificially lagged by any amount l by simply shifting the information from f_i to f_{i+l}. All of the components created are simulating a frequency spectrum, so all locations will be referred to as Hertz. The spectra have a frequency resolution of 0.01 Hz per frequency point.

When creating a data set, a set of random concentrations between 0.2 and 1 were created. If the data set was meant to be lagged, a maximum lag (max_lag) was defined to identify it, and a set of random lags between $[0, max\_lag]$ was created for that data set. To create the data set, (2) was applied.

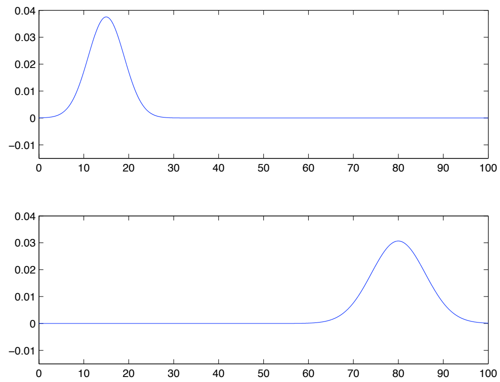
$$d_k = \sum_m c_m * \text{lag}(S_m, l_m) \quad (2)$$

where d_k is the kth spectrum in the data set; c_m and l_m are the randomly-generated concentration and lag, respectively, that are applied to the mth component in the reference spectra set S. The lag function shifts the information of the spectrum that is fed to it from f_i to f_{i+l_m}.

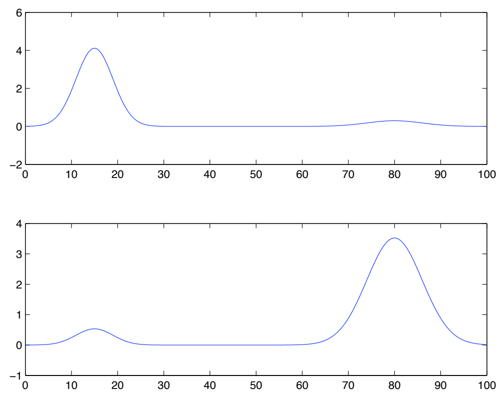
4 EXPERIMENTS & RESULTS

4.1 Experiment 1: Lag Effect on FastICA

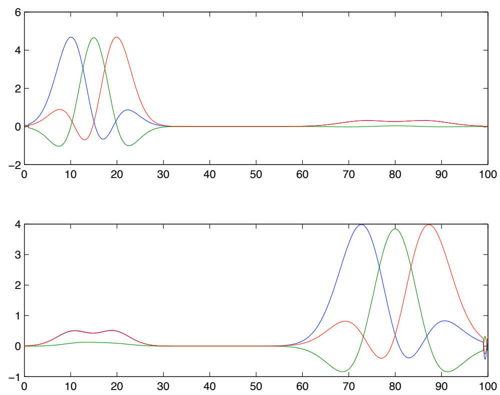
Two components were used: one at 15 Hz with a width of 20 Hz, and another at 80 Hz with a width of 30 Hz shown in Figure 1a. Two data sets were created using these components as reference spectra, having each 1000 signals using the same concentrations. However, one data set was lagged at a maximum of 1 Hz, while the other was not. The ICs identified from the non-lagged data set are shown in Figure 1b, and the ones identified from the lagged data set are shown in Figure 1c.



(a) Components used.



(b) ICs identified without lag.



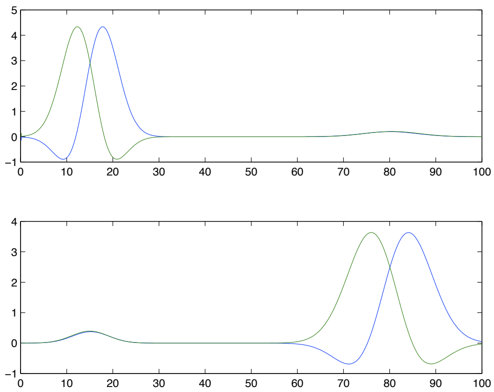
(c) ICs identifed with a max. lag of 1 Hz.

Figure 1: Results of first experiment.

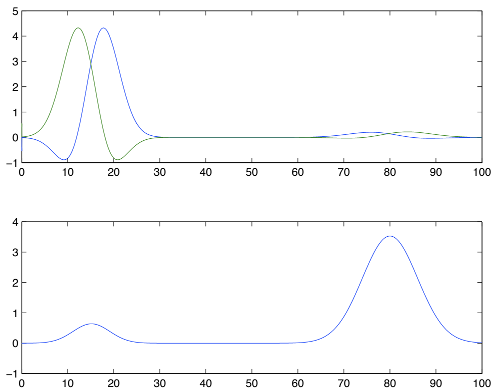
Figure 1c in reality is showing 6 components; they are grouped together like this for visualization sake.

4.2 Experiment 2: Maximum Lag Handled by FastICA

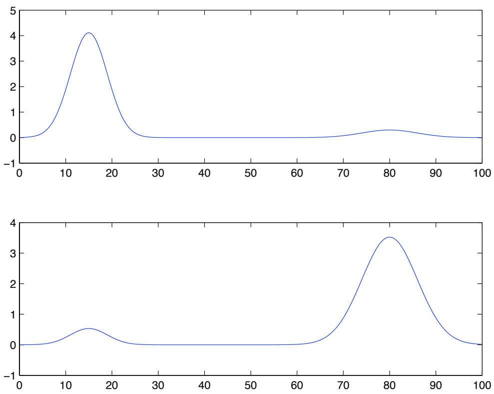
Using the same sources as in Experiment 1, different data sets were created, applying a different maximum lag to each one, ranging from 1 Hz to 0.01 Hz; FastICA was applied to all of them. Figure 2 shows, going from left to right, the ICs identified when applying a maximum lag of 0.1 Hz in Figure 2a, a maximum lag of 0.03 Hz in Figure 2b, and finally of 0.02 Hz in Figure 2c. Comparing the ICs shown in the latter to the ones shown in Figure 1b, that were obtained from non-lagged data, it is clear that FastICA is able to identify the correct ICs only applying a maximum lag of 0.02 Hz or less.



(a) ICs identifed with a max. lag of 0.1 Hz.



(b) ICs identifed with a max. lag of 0.03 Hz.

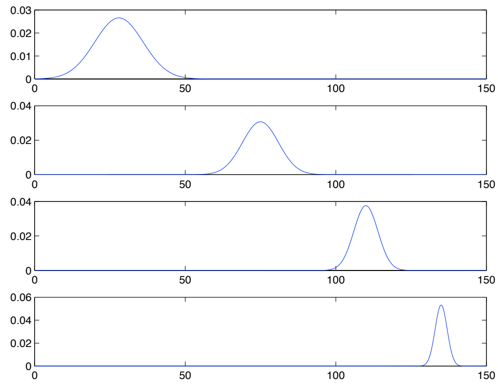


(c) ICs identified with a max. lag of 0.02 Hz.

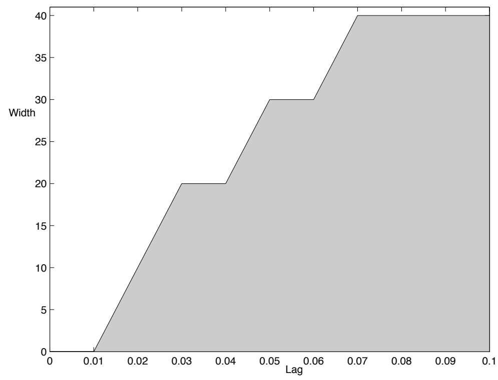
Figure 2: ICs obtained at different lags.

4.3 Experiment 3: Influence of Component Width

As it can be seen in Figure 2, when applying a maximum lag of 0.03 Hz, only the source with its peak at 80 Hz (which has a width of 30 Hz) was identified properly. To test if the proper identification of a lagged component is dependent of its width, another test was developed. Four sources were created and are shown in Figure 3a: each with a width of 40 Hz, 30 Hz, 20 Hz and 10 Hz, respectively. 10 data sets were created, each having 1000 signals, and a different maximum lag was applied to each one, from 0.01 Hz to 0.1 Hz; FastICA was applied to each data set.



(a) Sources used.



(b) Lag plotted against width.

Figure 3: Sources and lag/width plot.

Figure 3b shows an area, the border of which was obtained by plotting the maximum lag applied to a data set against the width of the widest component that was not properly identified. Any point residing in this area is describing a situation in which FastICA will *not* identify a component properly.

5 CONCLUSIONS & DISCUSSION

It has been shown that if a component suffers from moderate amounts of lag (which in real life, can be caused by sensor de-calibration), Independent Component Analysis and, more precisely, the FastICA algorithm, is unable to extract the same components that it does from non-lagged data. No other changes to the data, other than lagging it, was implemented, making the presence of lag the only responsible for making FastICA not identify the proper ICs.

The results in Experiments 2 and 3 show, however, that FastICA was able to properly identify the ICs in certain circumstances; basically, if the lag was small enough, and the component peaks were wide enough. As in other machine learning and advanced statistical methods, ICA considers each frequency point of a spectrum as if it were a variable, the values of which and their variation throughout the data set are key at extracting the needed information (in this case, the ICs). When lagging it, though, the information of one variable is passed onto others, so ICA is not able to follow the correct variation of the values of one variable, as they are now spread out into several. However, if the lag is small enough, the information is passed on to very close by neighbours of each variable, and, if the peaks are wide enough, the variation of their values are very similar, giving ICA almost the same information as if the data would not have suffered from any lag.

Nonetheless, it is important to note that it is still a *very small* amount of lag for such wide peaks for FastICA to identify the proper ICs. A lag of 0.07 Hz for a 40 Hz wide peak can be considered insignificant in most applications. This proves that ICA, particularly the FastICA implementation, is very fragile towards frequency displacement.

References

[1] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314, April 1994.

[2] C. Ladroue, F. A. Howe, J. R. Griffiths, and A. R. Tate. Independent component analysis for automated decomposition of in vivo magnetic resonance spectra. *Magnetic Resonance in Medicine*, 50(4):697–703, October 2003.

[3] H. Malika, A. Khokhar, and R. Ansari. Improved watermark detection for spread-spectrum based watermarking using independent component analysis. *Proceedings of the 5th ACM workshop on Digital rights management*, pages 102–111, 2005.

[4] M. Zuppa, C. Distante, K. C. Persaud, and P. Siciliano. Recovery of drifting sensor responses by means of dwt analysis. *Sensors & Actuators: B. Chemical*, 120(2):411–416, January 2007.

[5] Z.-P. Chen, J. Morris, and E. Martin. Correction of temperature-induced spectral variations by loading space standardization. *Analytical Chemistry*, 77(5):1376 – 1384, March 2005.

[6] K. Ichige, M. Imai, and H. Arai. Fastica-based blind signal separation and its applications to radio surveillance. *14th Workshop on Statistical Signal Processing*, pages 546–550, August 2007.

[7] Y. Tie and M. Sahin. Separation of spinal cord motor signals using the fastica method. *Neural Engineering*, 2:90–96, 2005.

[8] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. New York ; Chichester : Wiley, 2001.

[9] S. Roberts and R. Everson. *Independent component analysis : principles and practice*. Cambridge University Press, 2001.

[10] J. V. Stone. *Independent component analysis : a tutorial introduction*. Cambridge, Mass. ; London : MIT Press, 2004.

[11] A. Hyvärinen. New approximations of differential entropy for independent component analysis and projection pursuit. *Neural Computation*, 9(7):1483–1492, 1997.

[12] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

[13] A. Hyvärinen and E. Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4-5):411–430, May-June 2000.