Blind deconvolution with IIR filter
by projecting the learning law in FIR approximation

Takuya Hirotani, Norihiko Saito, Kenji Sugimoto
Nara Institute of Science and Technology
8916-5 Tayakama-cho, Ikoma-city, 630-0192, Japan
E-mail: takuya-hi@is.naist.jp, kenji@is.naist.jp

Abstract

Blind signal separation based on statistical independence has been studied extensively. Particularly difficult is separation of convolutive mixture. This can be solved approximately by a method with an FIR filter which is a cascade connection of many delay elements. It is, however, more desirable if the signal can be deconvoluted with an IIR filter (having feedback loop) of a modest order. This paper proposes such a demixer by projecting the conventional learning law based on the FIR approximation toward the parameter space of IIR filters. Numerical simulation is carried out to evaluate the proposed method.

1 Introduction

Recently, blind signal separation based on statistical independence has been studied extensively in the area of signal processing [1, 2]. The problem is to recover vector-valued discrete-time signal \( s(t) \) from observation \( y(t) = G * s(t), t = 1, \ldots, T \), where both \( G \) and \( s \) are assumed to be unknown, under certain assumptions; for a detail see e. g., [1, 2]. If, in particular, \( G \) is a constant square matrix, then \( * \) means mere multiplication and components of \( y \) are linear combinations of those of \( s \). In other words, \( y \) is an (instantaneous) mixture of \( s \). This is why the term “separation” is used for recovery of \( s \). Today’s research, however, mainly targets the case where \( G \) stands for discrete-time convolution, which is naturally far more difficult than the instantaneous mixture case and is called Blind Signal Deconvolution (BSD) shown in Fig. 1.

In \( z \)-domain, the above equation becomes \( y = G(z)s \) with a slight abuse of notation. BSD can be attained if we construct a filter playing a role of \( G(z)^{-1} \) (the filter is called demixer). A practical and well-known method for blind deconvolution is to approximate it by an FIR filter with sufficiently long taps [4]. In order to get enough precision, however, it requires a large amount of samples, and long calculation time.

The objective of this paper is to construct a demixer in a more direct way. To the observation we connect an IIR filter (i.e., having a backward as well as forward loops) of a suitable order and adjust its coefficients so that the processed signals are independent, thereby recovering the source signal in a sense. The adjusting law is given by projecting the learning law in FIR approximation as above.

In the next section we start by formulating our problem, and review the conventional method based on FIR approximation. Then in §3 we propose a new method, where we set an IIR filter of a given modest order and update the parameter. The effectiveness of the method is evaluated by means of simulation in §4.

Notations: \( \mathbb{R} \) is the set of real numbers. \( \text{diag}() \) stands for a diagonal matrix. \( I_m \) denotes the \( m \)-th order identity matrix.

2 Preliminaries

2.1 Problem formulation

Consider the source signal \( s(t) \in \mathbb{R}^m \), which is a vector-valued function of discrete-time \( t \). We assume that the components of \( s(t) \) are unknown but are random variables having the following properties:

- zero mean;
- spacially and temporally independent;
- identically distributed; and
- at most one of them is Gaussian.

Suppose that we observe the output signal of the system

\[
y(t) = G * s(t),
\]

where * denotes discrete-time convolution. The \( z \)-transformation of \( G \) is denoted by \( G(z) \), with the same symbol, which is assumed to be again unknown but a square (hence \( y \in \mathbb{R}^m \)), stable, and rational transfer matrix of minimal phase (hence its inverse is also a stable rational matrix).

Our objective is to recover the source signal by observing \( y(t) \) for a certain period (hence this is a batch processing), in the following sense: We connect a tunable filter \( H(z) \) which generates \( \hat{s} \in \mathbb{R}^m \), and we adjust \( H(z) \) so that the components of \( \hat{s}(t) \) are as independent as possible (a specific criterion will be given later).
Then it is known that

\[ H(z)G(z) = PA(z), \]
\[ A(z) = \text{diag}(\alpha_1 z^{-\nu_1}, \ldots, \alpha_m z^{-\nu_m}) \]

for some permutation matrix \( P \), scalars \( \alpha_i \), and integers \( \nu_i, i = 1, \ldots, m \). This means that the resulting \( \hat{s} \) is a recovery of \( s \) within indeterminacy of scalar multiplication, permutation, and time lag.

Fig. 1: Problem of blind source deconvolution

### 2.2 Conventional method: FIR approximation

In this subsection, we introduce a method to construct the demixer \( \hat{H}(z) \) which we formulate in the previous subsection. In this method we first consider the state space representation

\[ x(t+1) = Ax(t) + By(t), \quad \hat{s}(t) = Cx(t) + Dy(t), \]

where

\[ A = \begin{bmatrix} O_m & \cdots & O_m & O_m \\ O_m & I_m & \vdots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & I_m & O_m \end{bmatrix}, \quad B = \begin{bmatrix} I_m \\ O_m \\ \vdots \\ O_m \end{bmatrix} \]

\[ C = [H_1, H_2, \cdots, H_l], \quad D = H_0. \]

Note that \( A \) and \( B \) have the special structure, and hence \( H(z) = C(zI - A)^{-1}B + D \) is an FIR filter. In what follows, we denote the demixer given by the FIR method as \( H^F \), to distinguish it with general filters.

On the other hand \( C \) and \( D \) have nontrivial block elements which will be adjusted.

Fig. 2 is a block diagram of this FIR filter, where \( z^{-1} \) denotes the one step delay and \( l \) is the number of taps. We adjust parameters \( H_i^F(i = 0, 1, \cdots, l) \) so that components of \( \hat{s} \) become independent. Based on the Kullback-Leibler Divergence of \( \hat{s} \) and the above system representation, this is achieved by minimizing the cost function

\[ I(\hat{s}, H^F) = -\log |\det(D)| - \sum_{i=1}^n \log q_i(\hat{s}_i) \]

where \( H^F = [D, C] \), and \( q_i(\cdot) \) are the probability density functions of the source signals. Since they are unknown, we replace them with some suitable function as will be seen later.

\( I(\hat{s}, H^F) \) is non-negative and is equal to zero if and only if signals \( \hat{s} \) are mutually independent. This means that our objective is to find \( H^F \) which minimizes this cost.

It is, however, impossible to obtain the optimal solution analytically, hence we resort to a gradient descent algorithm. The update law based on the gradient of parameter is described by

\[ \Delta C(k) = -\eta(k)E[\varphi(\hat{s})x^T] \]
\[ \Delta D(k) = \eta(k)(I - E[\varphi(\hat{s})u^T]D(k)^TD(k) \]

where \( k \) denotes the repeating time and \( \eta(k) > 0 \) is a learning factor, and \( \varphi(y) \) is the score function. We take \( \varphi(y) = \tanh(y) \), if the distribution of the random variable \( y \) is super-Gaussian, while \( \varphi(y) = y^2 \), if \( y \) is sub-Gaussian. We skip a detail of the FIR approximation method; see e.g., [1, 2].

Fig. 2: Separation by FIR filter

### 3 Proposed Method

In this paper, we propose a method for updating IIR filter by projecting the learning law of FIR filter to parametric space of IIR filter (Fig. 3).

We assume transfer matrix of IIR filter as

\[ H^I(z) = D^{-1}(z)N(z) \]

where

\[ D(z) = I + D_1 z^{-1} + D_2 z^{-2} + \cdots + D_p z^{-p} \]
\[ N(z) = N_0 + N_1 z^{-1} + N_2 z^{-2} + \cdots + N_p z^{-p}. \]

Both \( p \) and \( q \) are known natural numbers. Since we assume small perturbation \( D(z) \to D(z) + \)
\[ \Delta D(z), N(z) \rightarrow N(z) + \Delta N(z), H^f(z) \rightarrow H^f(z) + \Delta H^f(z), \] and assign to (8), we obtain
\[ D(z) + \Delta D(z) \cdot H^f(z) = \Delta N(z). \] (11)

Now, we approximate \( H^f(z) \) by the FIR filter \( H^F(z) \), then we expand (11) in power series. By comparing its coefficient of \( z^{-i}, i = 0, 1, \ldots \), we can express
\[ (\Delta H^f_{i-1} + \Delta H^f_0 + \cdots + \Delta H^f_{p}) + (\Delta D_1 \cdot H^F_{i-1} + \cdots + \Delta D_p \cdot H^F_{i-1}) = \Delta N_i. \] (12)

Here, we define \( \Delta H^F_{i-1}, H^F_i \) are 0, if \( L < 0 \). Consider all of \( z^{-i} \), then we obtain
\[
[I, \tilde{D}] \begin{bmatrix}
\Delta H^F_0 & \Delta H^F_1 & \cdots & \Delta H^F_i & \cdots \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
\Delta H^F_0 & \Delta H^F_1 & \cdots & \Delta H^F_i & \cdots
\end{bmatrix}
+ \begin{bmatrix}
O & H^F_0 & H^F_1 & \cdots & H^F_i & \cdots \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
O & H^F_0 & H^F_1 & \cdots & H^F_i & \cdots
\end{bmatrix} = [\Delta \tilde{N}, O]. \] (13)

Here \( \Delta \tilde{D} = [\Delta D_1, \ldots, \Delta D_p], \tilde{D} = [D_1, \ldots, D_p], \Delta \tilde{N} = [\Delta N_0, \ldots, \Delta N_q], \tilde{N} = [N_0, \ldots, N_p]. \)

In the right-hand side, \((q+1)\)th column or later is zero matrix, so that we can separate (13) and get
\[
[I, \tilde{D}] \Delta \mathcal{H}_{\text{before}(q+1)} + \Delta \tilde{D} \cdot \mathcal{H}_{\text{before}(q+1)} = \Delta \tilde{N} \] (14)
\[
[I, \tilde{D}] \Delta \mathcal{H}_{\text{after}(q+1)} + \Delta \tilde{D} \cdot \mathcal{H}_{\text{after}(q+1)} = 0 \] (15)

Then we obtain \( \Delta D \) and \( \Delta N \) by solving (14) and (15) and can construct an inverse system as IIR filter by repeating this update law.

**Summary of Algorithm**

1. Initialize coefficients of \( D(z) \) and \( N(z) \): say \( D(z) = I_m \) and \( N(z) = I_m \).
2. Calculate \( H^F(z) \) (Approximate FIR filter) from current \( D(z) \) and \( N(z) \).
3. Calculate \( \Delta H^F(z) \) by the learning law in FIR approximation.
4. Calculate \( \Delta D(z) \) and \( \Delta N(z) \) from current \( D(z), N(z), H^F(z), \Delta H^F(z) \) by (14),(15).
5. Update \( D(z) \rightarrow D(z) + \Delta D(z), N(z) \rightarrow N(z) + \Delta N(z) \) (which means to update \( H^f \)).
6. If \( \Delta D(z) \) and \( \Delta N(z) \) are sufficiently small, then stop learning. Otherwise go back to Item 2.

4 Numerical simulation

In this section, we compare proposed method with conventional FIR approximation method[4]. In all conditions, the number of signals is assumed to be \( m = 2 \), and source signals and mixer are not used for deconvolution. Iteration is done until the error becomes sufficiently small, but maximum of iteration is 2000 in all cases. As we have pointed out, BSD has certain indeterminacy. For fair comparison, we eliminate it by hand.

4.1 A typical example

Consider following transfer function matrix
\[ G(z) = D^{-1}(z)N(z) \] (16)
\[ \tilde{D}(z) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} -0.29 & 0.60 \\ 0.95 & -0.56 \end{bmatrix} z^{-1} \] (17)
\[ \tilde{N}(z) = \begin{bmatrix} 0.79 & 0.16 \\ 0.36 & 0.48 \end{bmatrix} + \begin{bmatrix} 0.47 & 0.31 \\ 0.15 & 0.04 \end{bmatrix} z^{-1} \] (18)

which is stable and minimum phase, and whose impulse response and pole-zero map are given in Fig. 4.

![Impulse response](image1)

![Pole-zero map](image2)

Fig. 4: Property of the typical example mixer

Source signals \( s \) are binarized uniform random numbers (-1, or 1) shown in Fig. 5(a). Then we obtain
observed signals

\[ y = G(z)s \]  \hspace{1cm} (19)

shown in Fig. 5(b).

![Source signals (from 1 to 100)](image1)

(a)Source signals (from 1 to 100)

![Observed signals (from 1 to 100)](image2)

(b)Observed signals (from 1 to 100)

Fig. 5: Source signals and observed signals

\[ H^I(z) = D^{-1}(z)N(z) \]  \hspace{1cm} (20)

\[ D(z) = I + D_1z^{-1} \]  \hspace{1cm} (21)

\[ N(z) = N_0 + N_1z^{-1} \]  \hspace{1cm} (22)

For projection, we set also 20 taps coefficients of FIR approximation. In this condition, the number of iteration is 846 in conventional method, 391 in proposed method (but the proposed method takes about 50 percent longer computational time for one step than the conventional method). Fig. 6 shows the results by the proposed and conventional blind deconvolution methods.

Fig. 6 gives recovered signals, respectively by the conventional method (a), and by the proposed method. Simulation has been carried out for 1000 samples, but we show only signals between 1 to 100 in all figures in order to save space. Here, it would be ideal if we could evaluate the cost, since the learning law (14), (15) has been derived in order to minimize it. As is often the case in ICA, however, our scheme gives no explicit values of the cost.

Instead, we evaluate the error between source signals and recovered signals. Fig. 7 shows the error and it is clear that the proposed method has smaller recover error. Average of the absolute value of the recover error is respectively 0.0394 in the conventional method, 0.0139 in the proposed method.

![Error between source and recovered signals](image3)

Fig. 7: Error between source and recovered signals (blue: conventional, green: proposed)

We compare the accuracy of recovering system also. In system control engineering, BSD can be used for blind system identification and this is also our objective. Fig. 8 is bode diagram of true system and recovered system made by conventional and proposed method.

![Bode diagram](image4)

Fig. 8: Bode diagram (red: true, blue: conventional, green: proposed)
4.2 Statistical evaluation

An advantage of the proposed method is that it reduces the number of parameter, thereby leading to more effective learning. Hence we expect that the number of samples is reduced and the convergence is fast. In this subsection, we test the demixers with 10 different mixers and 4 different sample numbers. System parameters are chosen randomly, and for each of them we prepare 125, 250, 500, and 1000 samples for source signals. In choosing each mixer we require that it is stable and minimum phase, its impulse response is not so long, and it does not have a zero which is close to the unit circle.

Fig. 9 shows the relation between number of sample and average of recover error. Circle lied on middle of bar means average of 10 times and bar means standard deviate. Abscissa axis is the number of sample.

![Fig. 9: Relation between number of sample and recover error](image)

Fig. 9 shows the relation between number of sample and average of recover error. Circle lied on middle of bar means average of 10 times and bar means standard deviate. Abscissa axis is the number of sample.

Fig. 9 shows the relation between number of sample and average of recover error. Circle lied on middle of bar means average of 10 times and bar means standard deviate. Abscissa axis is the number of sample.

Fig. 10 indicates the relationship between the number of samples and the number of iteration. Bar and circle denote same as above. In this figure, as the sample number increases, the number of iteration increases in the conventional method, while it decreases in the proposed method.

4.3 Failure case

In this subsection, we show some cases where BSD failed, and consider its reason. Let us first consider the case where the mixer has the impulse response shown by Fig. 11. In this case, we can see that the impulse response of the mixer is much longer than the given tap number for FIR approximation. Hence, both conventional and proposed method couldn’t recover the signals and system in this case.

![Fig. 11: impulse response of first failed case](image)

Fig. 11: impulse response of first failed case

The second case is a mixer whose pole-zero map is given in Fig. 12. In this figure, we can see one of the zeros is close to the unit circle. In the proposed method, we resort to a numerous number of updating to obtain IIR filter, hence its stability is not guaranteed. The authors consider that the zero close to the unit circle has caused instability of the algorithm.

![Fig. 12: pole-zero mapping second failed case](image)

Fig. 12: pole-zero mapping second failed case
Table 1: simulation result of second failed case

<table>
<thead>
<tr>
<th>number of sample</th>
<th>125</th>
<th>250</th>
<th>500</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>method</td>
<td>conv</td>
<td>prop</td>
<td>conv</td>
<td>prop</td>
</tr>
<tr>
<td>times</td>
<td>141</td>
<td>448</td>
<td>1076</td>
<td>2000</td>
</tr>
<tr>
<td>error</td>
<td>0.1946</td>
<td>0.0992</td>
<td>0.1304</td>
<td>0.1496</td>
</tr>
</tbody>
</table>

5 Conclusions

In this paper, we proposed blind deconvolution by projecting the conventional learning law based on the FIR approximation toward the parameter space of IIR filters. By simulation results we see that the proposed method requires 50 percent longer calculation time for one step than the conventional one, due to high calculation cost for projection. Even so, as number of sample become larger, the proposed method converges faster, and recover accuracy is almost always better than conventional method in particular when we have few samples. On the other hand, the proposed method has turned out to be unstable in some cases.

In future work, we should explain the gradient of proposed method analytically, and make sure why proposed method is efficient.

References


