INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTÈMES ALÉATOIRES

PUBLICATION INTERNE
N° 1706

BSS_EVAL TOOLBOX USER GUIDE
REVISION 2.0

DEVELOPED WITH THE SUPPORT OF THE FRENCH GDR-ISIS/CNRS WORKGROUP "RESOURCES FOR AUDIO SOURCE SEPARATION" BY

C. FÉVOTTE , R. GRIBONVAL , E. VINCENT

ISSN 1166-8687

IRISA

CAMPUS UNIVERSITAIRE DE BEAULIEU - 35042 RENNES CEDEX - FRANCE

INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTÈMES ALÉATOIRES
Campus de Beaulieu – 35042 Rennes Cedex – France
Tél. : (33) 02 99 84 71 00 – Fax : (33) 02 99 84 71 71
http://www.irisa.fr

BSS_EVAL Toolbox User Guide
Revision 2.0

Developed with the support of the French GdR-ISIS/CNRS

Workgroup "Resources for Audio Source Separation" by

C. Févotte *, R. Gribonval **, E. Vincent ***

Systèmes cognitifs
Projet Metiss

Publication interne n° 1706 — April 2005 — 19 pages

Abstract: This document is meant to help you use the BSS_EVAL toolbox, which implements some criteria for performance measurement in (blind) source separation. The toolbox – which is distributed under the terms of the GNU GENERAL PUBLIC LICENSE as a set of MATLAB® **** routines – can be downloaded at the address `http://www.irisa.fr/metiss/bss_eval/`. The purpose of this toolbox is to measure the performance of various source separation algorithms in an evaluation framework where the original sources, and perhaps even the noise that perturbed the mixture, are available for comparison.

Key-words: source separation, performance measure, evaluation, source to distortion ratio, sources to interferences ratio, sources to noise ratio, sources to artifacts ratio, free software, user guide

(Résumé : tsvp)

* cf269@cam.ac.uk,
Dept. of Engineering, University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, UK.
** remi.gribonval@irisa.fr
*** emmanuel.vincent@elec.qmul.ac.uk
Centre for Digital Music, Queen Mary, University of London, Mile End Road London E1 4NS, UK.
**** MATLAB® is a registered trademark of The MathWorks, Inc.

Centre National de la Recherche Scientifique
(UMR 6074) Université de Rennes 1 – Insa de Rennes

Institut National de Recherche en Informatique
et en Automatique – unité de recherche de Rennes

Guide de l'utilisateur de la boîte à outils BSS_EVAL

Résumé : Ce document a pour objectif de vous aider à prendre en main et à utiliser la boîte à outils BSS_EVAL, qui met en œuvre quelques critères de mesure de performance pour la séparation (aveugle) de sources. Cette boîte à outils – distribuée sous les termes de la licence GNU GENERAL PUBLIC LICENSE en tant qu'ensemble de fonctions MATLAB® – peut être téléchargée à l'adresse http://www.irisa.fr/metiss/bss_eval/. L'objectif de cette boîte à outils est de mesurer la performance de divers algorithmes de séparation de sources dans un cadre d'évaluation où les sources d'origine, ainsi qu'éventuellement le bruit ajouté au mélange, sont disponibles pour servir de référence.

Mots clés : séparation de sources, mesure de performance, évaluation, rapport source à distortion, rapport sources à interférences, rapport sources à bruit, rapport sources à artefacts, logiciel libre, guide de l'utilisateur

Contents

Chapter 1

Getting started

This document is meant to help you use the `BSS_EVAL` toolbox, which implements the criteria for performance measurement in (blind) source separation described in the papers [1, 2].

1.1 License - no warranty

The toolbox is distributed under the terms of the GNU GENERAL PUBLIC LICENSE as a set of MATLAB® [1] routines so you should first get familiar with MATLAB® to use it.

1.2 Cite this as:

Within the limits of the GNU GENERAL PUBLIC LICENSE, you can use the toolbox as you please. If you use the toolbox in a work of your own that you wish to publish, please cite this user manual [3] including the URL of the toolbox

- C. Févotte, R. Gribonval and E. Vincent, *BSS_EVAL Toolbox User Guide*, IRISA Technical Report 1706, Rennes, France, April 2005. `http://www.irisa.fr/metiss/bss_eval/`.

1.3 Download and install

The latest version of the toolbox can be downloaded at `http://www.irisa.fr/metiss/bss_eval/`. Once you have downloaded and uncompressed the toolbox you should get the following toolbox files

- LICENSE.txt
- Contents.m
- bss_crit.m
- bss_decomp_gain.m
- bss_decomp_filt.m

[1] MATLAB® is a registered trademark of The MathWorks, Inc.

- bss_decomp_tvgain.m
- bss_decomp_tvfilt.m
- bss_proj.m
- bss_tvproj.m
- bss_make_frames.m
- bss_make_lags.m
- bss_energy_ratios.m

as well as the present documentation in PostScript and PDF: `user_guide.ps`, `user_guide.pdf`.

1.4 Getting help

Within MATLAB® , from the directory where the toolbox files are located, you can get basic online help on the various functions of the toolbox by typing

```
help Contents
```

If you have added the toolbox directory to the MATLAB® path you can simply type

```
help BSS_EVAL
```

1.5 Reading guide

In chapter 2 you will learn how to use the various functions of the toolbox to compute performance measures for source separation. Chapter 3 gives a detailed documentation for each function of the toolbox.

Chapter 2

User guide

2.1 Context

The purpose of this toolbox is to measure the performance of various source separation algorithms in an evaluation framework where the original sources, and perhaps even the noise that perturbed the mixture, are available for comparison.

2.2 Principle

The principle of the performance measures described in [1] is to **decompose** a given estimate $\widehat{s}(t)$ of a source $s_i(t)$ as a sum

$$\widehat{s}(t) = s_{\text{target}}(t) + e_{\text{interf}}(t) + e_{\text{noise}}(t) + e_{\text{artif}}(t) \tag{2.1}$$

where $s_{\text{target}}(t)$ is an allowed deformation of the target source $s_i(t)$, $e_{\text{interf}}(t)$ is an allowed deformation of the sources which accounts for the interferences of the unwanted sources, $e_{\text{noise}}(t)$ is an allowed deformation of the perturbating noise (but not the sources), and $e_{\text{artif}}(t)$ is an "artifact" term that may correspond to artifacts of the separation algorithm such as musical noise, etc. or simply to deformations induced by the separation algorithm that are not allowed. There are several ways of computing such a decomposition depending on which transformations are allowed, and the toolbox includes the function `bss_decomp_gain` (resp. `bss_decomp_filt`, `bss_decomp_tvgain`, `bss_decomp_tvfilt`) to perform the decomposition when the allowed deformation is a constant gain (resp. a constant filter, a time-varying gain, a time-varying filter). As an example, you would type

$$[\texttt{s\_target}, \texttt{e\_interf}, \texttt{e\_noise}, \texttt{e\_artif}] = \texttt{bss\_decomp\_gain}(\texttt{se}, \texttt{i}, \texttt{S}, \texttt{N}) \tag{2.2}$$

to get the decomposition of the estimated source `se` (a row vector in MATLAB®) with `S(i,:)` the target source, `S(j,:)`, $\texttt{j} \neq \texttt{i}$ (the rows of the matrix `S`) the other sources, and `N(k,:)` the perturbating noise signals. In the case where the noise signals are not known or there is no noise, you would simply type

$$[\texttt{s\_target}, \texttt{e\_interf}, \texttt{e\_artif}] = \texttt{bss\_decomp\_gain}(\texttt{se}, \texttt{i}, \texttt{S}) \tag{2.3}$$

Given such a decomposition, one can compute performance **criteria** either globally or, as we will see later, locally. Four global performance measures are defined : the Source to Distortion

Ratio

$$\text{SDR} := 10 \log_{10} \frac{\|s_{\text{target}}\|^2}{\|e_{\text{interf}} + e_{\text{noise}} + e_{\text{artif}}\|^2}, \tag{2.4}$$

the Source to Interferences Ratio

$$\text{SIR} := 10 \log_{10} \frac{\|s_{\text{target}}\|^2}{\|e_{\text{interf}}\|^2}, \tag{2.5}$$

the Sources to Noise Ratio

$$\text{SNR} := 10 \log_{10} \frac{\|s_{\text{target}} + e_{\text{interf}}\|^2}{\|e_{\text{noise}}\|^2}, \tag{2.6}$$

and the Sources to Artifacts Ratio

$$\text{SAR} := 10 \log_{10} \frac{\|s_{\text{target}} + e_{\text{interf}} + e_{\text{noise}}\|^2}{\|e_{\text{artif}}\|^2}. \tag{2.7}$$

To compute these criteria, simply type

$$[\texttt{SDR}, \texttt{SIR}, \texttt{SNR}, \texttt{SAR}] = \texttt{bss\_crit}(\texttt{s\_target}, \texttt{e\_interf}, \texttt{e\_noise}, \texttt{e\_artif}) \tag{2.8}$$

In case `e_noise` has not been computed, you can also type

$$[\texttt{SDR}, \texttt{SIR}, \texttt{SAR}] = \texttt{bss\_crit}(\texttt{s\_target}, \texttt{e\_interf}, \texttt{e\_artif}) \tag{2.9}$$

2.3 Global *vs* local criteria

Sometimes, it is not very satisfying to summarize the performance by a single figure for the whole signal: it may happen that on some pieces of the estimated signal the interferences are very low because the target source is loud, but on other pieces the target source vanishes. To obtain local performance measures we provide the syntax

$$[\texttt{SDR}, \texttt{SIR}, \texttt{SNR}, \texttt{SAR}] = \texttt{bss\_crit}(\texttt{s\_target}, \texttt{e\_interf}, \texttt{e\_noise}, \texttt{e\_artif}, \texttt{window}, \texttt{NOVERLAP}) \tag{2.10}$$

where the meaning of `window` and `NOVERLAP` is the same as in the `specgram` function of MATLAB® Signal Processing Toolbox. It is not important whether the window is smooth (its only effect is to compute *local* performance measures, but no FFT is performed, no side-lobes are to be feared, etc.) so you can safely use rectangular windows such as `ones(1,512)`. With this syntax, instead of being a number, `SDR` (resp. `SIR`, `SNR` and `SDR`) is a column vector which entries `SDR(n)` correspond to the local performance on the `n`-th frame, that is to say

$$\text{SDR}(n) := 10 \log_{10} \frac{\|w(t-t_n) \cdot s_{\text{target}}(t)\|^2}{\|w(t-t_n) \cdot (e_{\text{interf}}(t) + e_{\text{noise}}(t) + e_{\text{artif}}(t))\|^2}. \tag{2.11}$$

Use `plot(SDR)` to display the variations of these performance measures along the frames, or compute and display its cumulative histogram to get statistics on its values.

2.4 Multiple target sources – advanced usage

Estimating a single specific source $s_i(t)$ is only one of the many goals in "source separation": it happens that one may be more interested in recovering estimates of combinations of several sources. For example, in karaoke, $\widehat{s}(t)$ will be good if the voice source has been correctly rejected, but it does not really matter if the result is a reasonable deformation of the other sources. To measure the performance in such contexts, one can use the decomposition functions (`bss_decomp_gain`, etc.) as

$$[\texttt{s\_target}, \texttt{e\_interf}, \texttt{e\_noise}, \texttt{e\_artif}] = \texttt{bss\_decomp\_gain}(\texttt{se}, \texttt{index}, \texttt{S}, \texttt{N}) \qquad (2.12)$$

with `index` a column vector of indices indicating which sources (rows of `S`) are target sources. Thus, in the karaoke example, `index` would contain all the sources indexes except for the voice.

2.5 Diagnostic – listening to the artifacts

By examining (or listening to) the signals [`s_target`,`e_interf`,`e_noise`,`e_artif`], one can determine whether the chosen decomposition (that is to say the set of allowed distortions) is meaningful for the target at hand, in particular whether the notion of "artifacts" meets its intuitive meaning.

2.6 Time-varying decompositions

When using the decompositions designed to deal with time-varying gains or time-varying filters, you must specify a "shape" $v(t)$ of the variations and a step (in number of samples) which together determine which variations are considered admissible. Since the smoothness of $v(t)$ determines that of the allowed variations, you will probably want to avoid the rectangular window (unless you want to allow piecewise constant gains / piecewise constant filters). A good choice would probably be to use a triangular window (for piecewise linear variations) or higher order splines.

Bibliography

[1] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Trans. Speech and Audio Proc.*, 2005, to appear.

[2] R. Gribonval, L. Benaroya, E. Vincent, and C. Févotte, "Proposals for performance measurement in source separation," in *Proc. 4th Int. Symp. on Independent Component Anal. and Blind Signal Separation (ICA2003)*, Nara, Japan, Apr. 2003, pp. 763–768.

[3] C. Févotte, R. Gribonval, and E. Vincent, "BSS_EVAL toolbox user guide," IRISA, Rennes, France, Tech. Rep. 1706, 2005. [Online]. Available: `http://www.irisa.fr/metiss/bss_eval/`

Chapter 3

Reference manual

`bss_crit`

Purpose:

Computes evaluation criteria given a decomposition of an estimated source into target sources, interfering sources, perturbating noise and artifacts contributions.

Synopsis (global mode):

```
    [SDR,SIR,SAR]  =  bss_crit(s_target,e_interf,e_artif)
[SDR,SIR,SNR,SAR]  =  bss_crit(s_target,e_interf,e_noise,e_artif)
```

Synopsis (local mode):

```
    [SDR,SIR,SAR]  =  bss_crit(s_target,e_interf,e_artif,WINDOW,NOVERLAP)
[SDR,SIR,SNR,SAR]  =  bss_crit(s_target,e_interf,e_noise,e_artif,WINDOW,NOVERLAP)
```

Input:

Name	Description
`s_target`	contribution of the target source(s)
`e_interf`	contribution of interfering sources
`e_noise`	(optional) contribution of perturbating noise
`e_artif`	contribution of artifacts
`WINDOW`	row vector containing the window used in local mode
`NOVERLAP`	number of samples of overlap between adjacent windows

Output (global mode):

`SDR`, `SIR`, `SNR` and `SAR` are scalars

Output (local mode):

`SDR`, `SIR`, `SNR` and `SAR` are column vectors which entries correspond to the local performance on each frame, see Eq. (2.11).

`bss_decomp_gain`

Synopsis:

```
        [s_target,e_interf,e_artif]  =  bss_decomp_gain(se,index,S)
[s_target,e_interf,e_noise,e_artif]  =  bss_decomp_gain(se,index,S,N)
```

Description:

Decomposes an estimated source into the contributions of the target sources, of the interfering sources, of perturbating noise and of the rest named artifacts. The only allowed deformation is a **pure gain**, so when the input sources and noises are mutually orthogonal, the contributions are computed based on the model

$$\widehat{s}(t) = \sum_{i \in I} a_i \cdot s_i(t) + \sum_{j \notin I} a_j \cdot s_j(t) + \sum_k b_k \cdot n_k(t) + e_{\mathrm{artif}}(t). \tag{3.1}$$

See [1] or the file `bss_gain.m` to learn how the contributions are computed for non mutually orthogonal input.

Input:

Name	Description
`se`	row vector representing the estimated source $\widehat{s}(t)$
`index`	column vector of indices of the target sources in the rows of `S`, representing the set I
`S`	matrix which rows correspond to the original sources (target $s_i(t)$, $i \in I$ + interfering $s_j(t)$, $j \notin I$)
`N`	(optional) matrix which rows correspond to the perturbating noise signals $n_k(t)$

Output:

`s_target`, `e_interf`, `e_noise`, `e_artif` : row vectors of the same dimension as `se`.

`bss_decomp_filt`

Synopsis:

```
        [s_target,e_interf,e_artif]  =  bss_decomp_filt(se,index,S,L)
[s_target,e_interf,e_noise,e_artif]  =  bss_decomp_filt(se,index,S,N,L)
```

Description:

Decomposes an estimated source into the contributions of the target sources, of the interfering sources, of perturbating noise and of the rest named artifacts. The only allowed deformation is a **pure filter**, of controled tap length, so when the input sources and noises are mutually orthogonal the contributions are computed based on the model

$$\widehat{s}(t) = \sum_{i \in I} \sum_{l=0}^{L-1} a_i(l) \cdot s_i(t-l) + \sum_{j \notin I} \sum_{l=0}^{L-1} a_j(l) \cdot s_j(t-l) + \sum_{k} \sum_{l=0}^{L-1} b_k(l) \cdot n_k(t-l) + e_{\text{artif}}(t). \quad (3.2)$$

See [1] or the file `bss_filt.m` to learn how the contributions are computed for non mutually orthogonal input.

Input:

Name	Description
`se`	row vector representing the estimated source $\widehat{s}(t)$
`index`	column vector of indices of the target sources in the rows of `S`, representing the set I
`S`	matrix which rows correspond to the original sources (target $s_i(t)$, $i \in I$ + interfering $s_j(t)$, $j \notin I$)
`N`	(optional) matrix which rows correspond to the perturbating noise signals
`L`	number of taps allowed in the distorting filters

Output:

`s_target`, `e_interf`, `e_noise`, `e_artif` : row vectors of the same dimension as `se`.

`bss_decomp_tvgain`

Synopsis:

```
        [s_target,e_interf,e_artif]  =  bss_decomp_tvgain(se,index,S, tvshape, tvstep)
[s_target,e_interf,e_noise,e_artif]  =  bss_decomp_tvgain(se,index,S,N, tvshape, tvstep)
```

Description:

Decomposes an estimated source into the contributions of the target sources, of the interfering sources, of perturbating noise and of the rest named artifacts. The only allowed deformation is a **(slowly) time varying gain**, so when the input sources and noises are mutually orthogonal the contributions are computed based on the model

$$\widehat{s}(t) = \sum_{i \in I} a_i(t) s_i(t) + \sum_{j \notin I} a_j(t) s_j(t) + \sum_k b_k(t) n_k(t) + e_{\text{artif}}(t) \tag{3.3}$$

where the gains $a_i(t)$ (resp. $b_k(t)$) are slowly time-varying in the sense that they have the parametric form

$$a_i(t) = \sum_r \alpha_i(r) \cdot v(t - r \cdot T) \tag{3.4}$$

with $v(t)$ a smooth "window" and $T \gg 1$ a rate of variation. See [1] or the file `bss_tvgain.m` to learn how the contributions are computed for non mutually orthogonal input.

Input:

Name	Description
`se`	row vector representing the estimated source $\widehat{s}(t)$
`index`	column vector of indices of the target sources in the rows of `S`, representing the set I
`S`	matrix which rows correspond to the original sources (target $s_i(t)$, $i \in I$ + interfering $s_j(t)$, $j \notin I$)
`N`	(optional) matrix which rows correspond to the perturbating noise signals
`tvshape`	row vector containing the shape $v(t)$ of the variations of the gain
`tvstep`	number of samples T of distance between adjacent variations of the gain

Output:

`s_target`, `e_interf`, `e_noise`, `e_artif` : row vectors of the same dimension as `se`.

`bss_decomp_tvfilt`

Synopsis:

```
        [s_target,e_interf,e_artif]  =  bss_decomp_tvfilt(se,index,S,tvshape,tvstep,L)
 [s_target,e_interf,e_noise,e_artif]  =  bss_decomp_tvfilt(se,index,S,N,tvshape,tvstep,L)
```

Description:

Decomposes an estimated source into the contributions of the target sources, of the interfering sources, of perturbating noise and of the rest named artifacts. The only allowed deformation is a **(slowly) time varying filter**, so when the input sources and noises are mutually orthogonal the contributions are computed based on the model

$$\widehat{s}(t) = \sum_{i\in I}\sum_{l=0}^{L-1} a_i(l,t)\cdot s_i(t-l) + \sum_{j\notin I}\sum_{l=0}^{L-1} a_j(l,t)\cdot s_j(t-l) + \sum_{k}\sum_{l=0}^{L-1} b_k(l,t)\cdot n_k(t-l) + e_{\mathrm{artif}}(t) \quad (3.5)$$

where the filter coefficients $a_i(l,t)$ (resp. $b_k(l,t)$) vary slowly with time in the sense that they have the parametric form

$$a_i(l,t) = \sum_r \alpha_i(l,r)\cdot v(t - r\cdot T). \quad (3.6)$$

with $v(t)$ a smooth "window" and $T \gg 1$ a rate of variation. See [1] or the file `bss_tvfilt.m` to learn how the contributions are computed for non mutually orthogonal input.

Input:

Name	Description
`se`	row vector representing the estimated source $\widehat{s}(t)$
`index`	column vector of indices of the target sources in the rows of `S`, representing the set I
`S`	matrix which rows correspond to the original sources (target $s_i(t)$, $i \in I$ + interfering $s_j(t)$, $j \notin I$)
`N`	(optional) matrix which rows correspond to the perturbating noise signals
`tvshape`	row vector containing the shape $v(t)$ of the variations of the gain
`tvstep`	number of samples T of distance between adjacent variations of the gain
`L`	number of taps allowed in the distorting filters

Output:

`s_target`, `e_interf`, `e_noise`, `e_artif` : row vectors of the same dimension as `se`.

```
bss_proj
```

Synopsis:

```
        PY_x  =  bss_proj(x,Y)
[PY_x coeff]  =  bss_proj(x,Y)
```

Description:

Computes the orthogonal projection of a signal $x(t)$ onto the subspace spanned by other signals $y_i(t)$, that is to say

$$P_Y x(t) = \sum_i c_i y_i(t) \tag{3.7}$$

with $x - P_Y x$ orthogonal to each vector y_i.

Input:

Name	Description
`x`	row vector representing the signal $x(t)$
`Y`	matrix or row vector which rows correspond to the signals $y_i(t)$

Output:

Name	Description
`PY_x`	row vector representing the projected signal $P_Y x(t)$
`coeff`	column vector corresponding to the coefficients c_i

Remark:

The projection will not properly work if the rows of `Y` are linearly dependent (e.g., if two sources are identical).

```
bss_tvproj
```

Synopsis:

```
        PY_x   =  bss_tvproj(x,Y,tvshape,tvstep)
 [PY_x coeff]  =  bss_tvproj(x,Y,tvshape,tvstep)
```

Description:

Computes the orthogonal projection of a signal $x(t)$ onto the subspace spanned by the windowed versions of other signals $y_i(t)$, that is to say

$$P_Y x(t) = \sum_{i,r} c_{i,r} \cdot v(t - rT) \cdot y_i(t) \tag{3.8}$$

with $x(t) - P_Y x(t)$ orthogonal to each windowed vector $v(t - rT) \cdot y_i(t)$.

Input:

Name	Description
`x`	row vector representing the signal $x(t)$
`Y`	matrix or row vector which rows correspond to the signals $y_i(t)$
`tvshape`	row vector containing the shape $v(t)$ of the window
`tvstep`	number of samples T of distance between adjacent variations of the gain

Output:

Name	Description
`PY_x`	row vector representing the projected signal $P_Y x(t)$
`coeff`	matrix corresponding to the coefficients $c_{i,r}$ (rows correspond to rows of `Y`, columns to frames)

`bss_make_frames`

Synopsis:
```
[F_S frames_index] = bss_make_frames(S,WINDOW,NOVERLAP)
```

Description:

Decompose some signal(s) into frames

Input:

Name	Description
`S`	matrix of size $n \times T$ which rows correspond to the signals $y_i(t)$
`WINDOW`	row vector of size $1 \times W$ containing the window
`NOVERLAP`	number of samples of overlap between adjacent windows

Output:

Name	Description
`F_S`	$n_{frames} \times W \times n$ tensor containing the frames (of length W) of each row of S
`frames_index`	index of the beginning of each frame in the rows of `S`

Remark:
If $n = 1$, `F_S` is a matrix of size $n_{frames} \times W$

`bss_make_lags`

Synopsis:

```
S_lags = bss_make_lags(S,L)
```

Description:

Create a matrix containing lagged (delayed) versions of some signals.

Input:

Name	Description
`S`	matrix of size $n \times T$ which rows contain input signals $s_n(t)$
`L`	number of lagged versions of the signal(s)

Output:

Name	Description
`S_lagged`	matrix of size $(nL) \times T$ which rows represent the lagged signals

`bss_energy_ratios`

Purpose:

Computes energy ratios corresponding to SDR/SIR/SNR/SAR given a decomposition of an estimated source into target sources, interfering sources, perturbating noise and artifacts contributions.

Synopsis:

```
    [SDR,SIR,SAR]  =  bss_energy_ratios(F_s_target,F_e_interf,F_e_artif)
[SDR,SIR,SNR,SAR]  =  bss_energy_ratios(F_s_target,F_e_interf,F_e_noise,F_e_artif)
```

Input:

Name	Description
`F_s_target`	n_{frames} x T matrix containing the frames of the contribution of the target source(s)
`F_e_interf`	n_{frames} x T matrix containing the frames of the contribution of interfering sources
`F_e_noise`	(optional) n_{frames} x T matrix containing the frames of the contribution of perturbating noise
`F_e_artif`	n_{frames} x T matrix containing the frames of the contribution of artifacts

Output:

`SDR`, `SIR`, `SNR` and `SAR` are column vectors of size n_{frames} which entries correspond to the local performance on each frame, see Eq. (2.11).