

This Matlab/C++ program implements the cochannel speech separation algorithm described in

K. Hu and D. L. Wang (2013), "An iterative model-based approach to cochannel speech separation," EURASIP Journal on Audio, Speech, and Music Processing, vol. 2013, Article ID 2013-14 (11 pages).

The MATLAB program `run/twoSpeaker.m` is a wrapper including several related model-based algorithms. The core separation algorithms are written in C++ under folder "c".

Usage:

```
rmask = twoSpeaker(sig, sid, type, nGau, bW, snr_criterion, nStep, workFolder)
```

Inputs:

sig: Input time-domain cochannel speech signal

sid: Two speaker identities (sid(1) and sid(2))

type: 'acoustDym_iter' – The iterative model-based algorithm

'ReddyRaj' - Reddy & Raj'07 (training and test energy levels must match)

'MMSE' - Minimum mean square estimation

'MAP' - Maximum *a posteriori* estimation

'acoustDym' – With temporal dynamics

'MMSE_iter' – MMSE + iterative estimation

'MAP_iter' – MAP + iterative estimation

nGau: Number of Gaussians in GMM (use 256 in this code)

bW: Beam width in a Viterbi search (use 16; only used in HMM-based algorithms)

snr_criterion: A threshold (in dB) on the absolute SNR difference to stop iterative estimation (use 0.5)

nStep: Maximum number of iterations (make sure iterative estimation will stop)

workFolder: Folder storing temporary files

Outputs:

rmask: Estimated soft masks for two speaker (mask{1} and mask{2})

Run an example:

In Linux, go to the "run" folder, start MATLAB, and input the following commands:

```
sig = load('sample/t11_lwby6p_m30_lrwp2a.-9dB.val2') ;  
rmask = twoSpeaker(sig, [11,30], 'acoustDym_iter', 256, 16, .5, 3, '.');
```

Run related model-based methods:

```
sig = load('sample/t11_lwby6p_m30_lrwp2a.-9dB.val2') ;
```

- Reddy & Raj'07 (Training and test signal levels must match):

```
rmask = twoSpeaker(sig, [11,30], 'ReddyRaj07', 256, -1, -1, -1, '.');
```


 (-1 means no parameter needed)
- MMSE:

```
rmask = twoSpeaker(sig, [11,30], 'MMSE', 256, -1, -1, -1, '.');
```
- MAP:

```
rmask = twoSpeaker(sig, [11,30], 'MAP', 256, -1, -1, -1, '.');
```
- MAP + acoustic dynamics:

```
rmask = twoSpeaker(sig, [11,30], 'acoustDym', 256, 16, -1, -1, '.');
```
- MMSE + iterative:

```
rmask = twoSpeaker(sig, [11,30], 'MMSE_iter', 256, -1, .5, 3, '.');
```
- MAP + iterative:

```
rmask = twoSpeaker(sig, [11,30], 'MAP_iter', 256, -1, .5, 3, '.');
```

Notes:

- Speakers in this program come from the Speech Separation Challenge (SSC) corpus. Speaker identities are numbered from 1-34 following the definition in the SSC corpus. For example, sid=[1,2] means the target is speaker 1 and interferer is speaker 2
- This implementation is in the log-cochleagram domain using 128-channel gammatone filterbank
- Sampling frequency is 16 kHz