

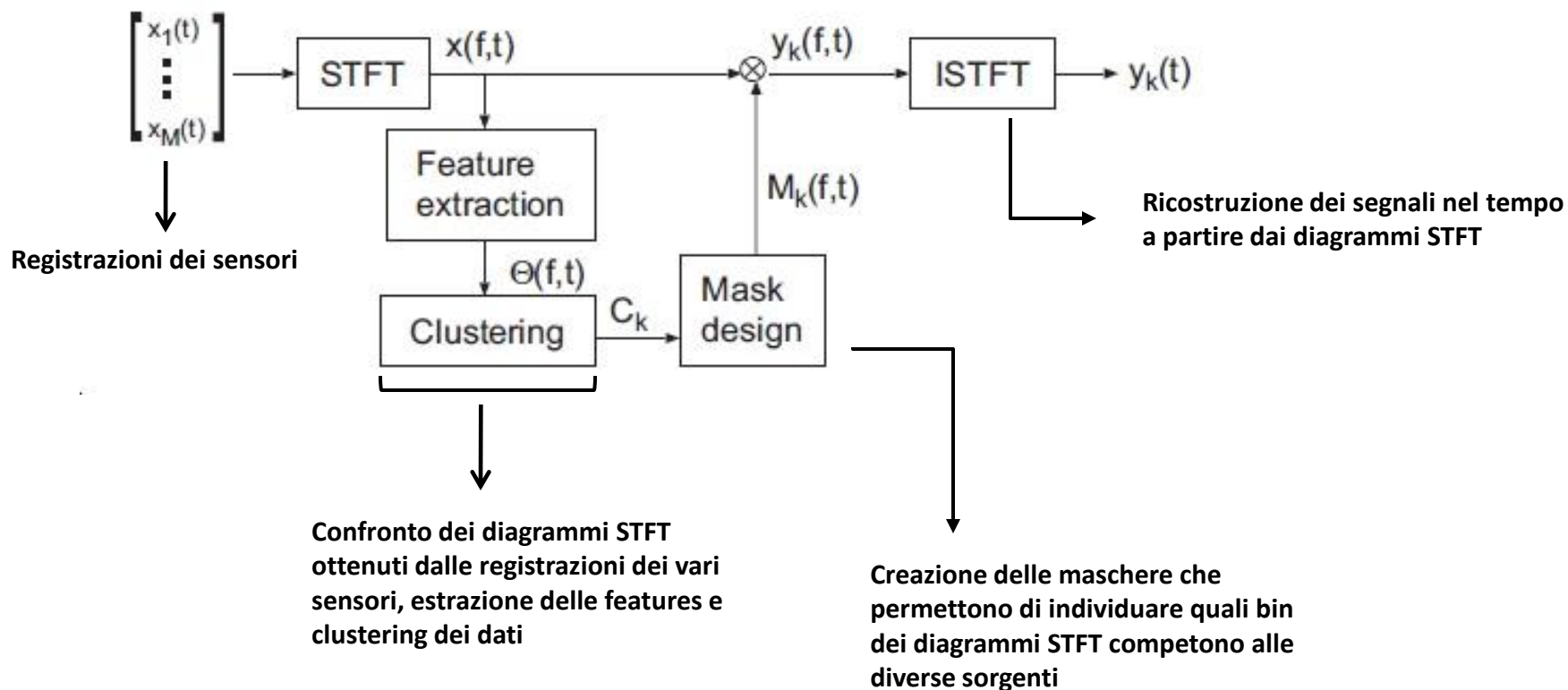
Blind Source Separation with binary **Mask approach**

Intento delle prove

- Implementare e verificare il funzionamento di un algoritmo di separazione di sorgenti audio basato su di un'analisi nel dominio delle trasformate STFT per un problema sottodeterminato (più sorgenti che sensori)
- Verificare le performance e le criticità dell'algoritmo

Descrizione dell'algoritmo

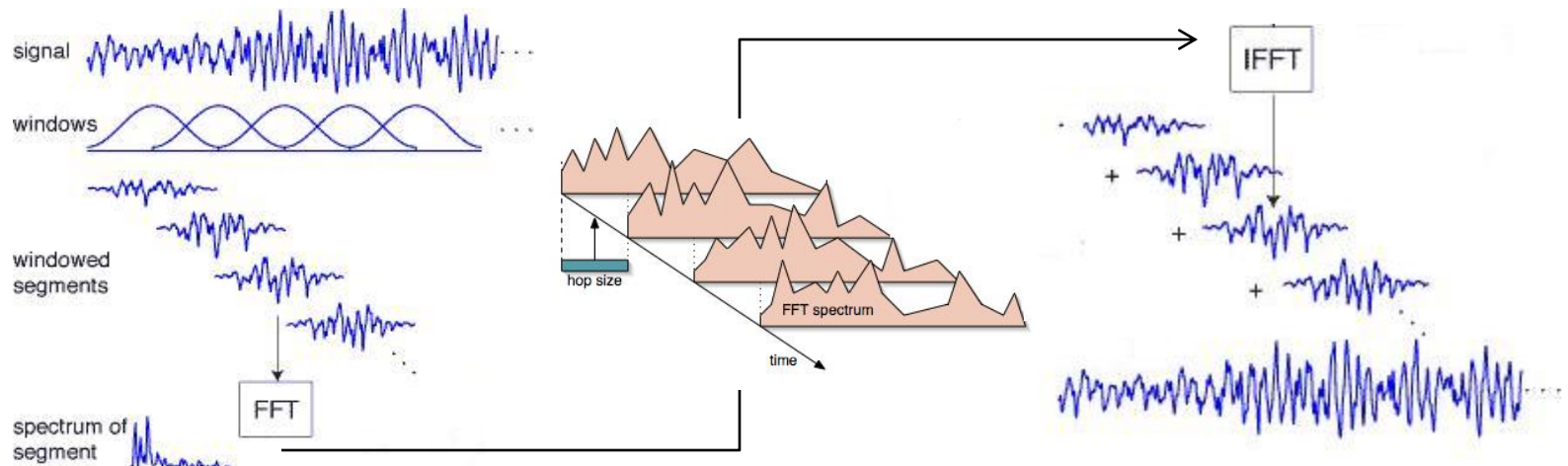
Si riporta uno schema della procedura di separazione che si è implementata:



STFT/ISTFT

→ Attraverso analisi STFT è possibile analizzare l'evoluzione temporale dello spettro di un segnale

→ La sparsità delle sorgenti nel dominio tempo-frequenza è molto più alta di quella riscontrabile nel dominio diretto del tempo o nel dominio trasformato delle frequenze



→ I migliori risultati si sono ottenuti utilizzando una finestra di hamming da $M=256$ campioni, con avanzamento $M/2$

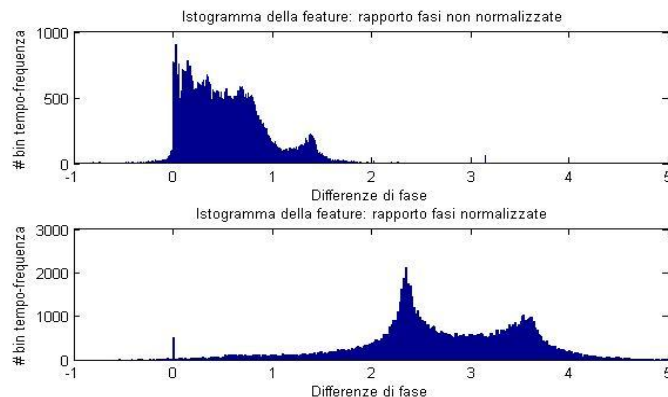
→ La ricostruzione tramite trasformata inversa ISTFT prevede l'applicazione del paradigma dell'OVERLAP & ADD nella scelta della finestra e dell'avanzamento

Feature extraction

→ Per determinare l'appartenenza di ogni bin del dominio tempo-frequenza ad una piuttosto che ad un'altra sorgente è necessario determinare delle caratteristiche (features), tali da permettere la classificazione dei punti del dominio

→ Le FEATURES utilizzate sono:

- Quoziente dei moduli dei diagrammi STFT delle registrazioni dei due microfoni
- Differenze di fase opportunamente normalizzate tra i punti dei diagrammi STFT



→ Non normalizzate

→ Normalizzate

Features:

$$\frac{|STFT1(f,t)|}{|STFT2(f,t)|}$$

$$\frac{1}{2\pi f c^{-1} d} \arg\left(\frac{STFT1(f,t)}{STFT2(f,t)}\right)$$

→ La scelta di queste features deriva dall'applicazione delle procedure indicate nell'articolo seguente: "Underdetermined blind sparse source separation for arbitrarily arranged multiple sensors"

Shoko Arakia, Hiroshi Sawada, Ryo Mukaia, Shoji Makinoa

Classificazione

→ I vettori delle features vengono utilizzati da uno specifico algoritmo che ne implementa la CLASSIFICAZIONE

→ Tra i vari algoritmi di classificazione esistenti si è scelto di utilizzare il K-MEANS

→ L'algoritmo K-means è un algoritmo di clustering che permette di suddividere gruppi di oggetti in K partizioni sulla base dei loro attributi, delle loro features

→ Ogni cluster viene identificato mediante un centroide o punto medio. L'algoritmo segue una procedura iterativa:

- Inizialmente crea K partizioni e assegna ad ogni partizione i punti d'ingresso o casualmente o usando alcune informazioni euristiche
- Calcola il centroide di ogni gruppo
- Costruisce quindi una nuova partizione associando ogni punto d'ingresso al cluster il cui centroide è più vicino ad esso
- Vengono ricalcolati i centroidi per i nuovi cluster

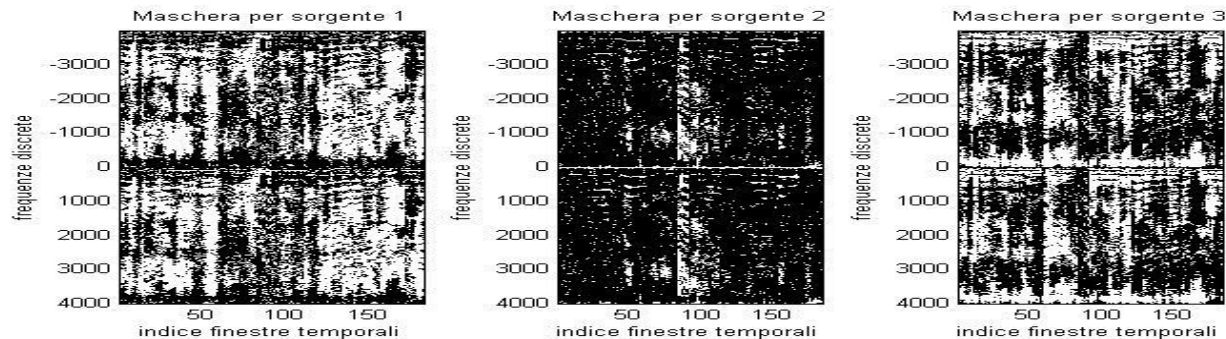
→ Per definire il concetto di vicinanza possono essere utilizzate varie metriche; nel nostro caso si è scelto di utilizzare una metrica in NORMA L1 per proteggersi da OUTLIER nei vettori delle feature

Maschere binarie/Ricostruzione

→ Una volta effettuata la classificazione si hanno a disposizione le informazioni necessarie per associare ogni bin dei diagrammi STFT alla sorgente a cui compete

→ Si creano le cosiddette MASCHERE BINARIE, una per ogni sorgente, nel seguente modo:

- per ogni sorgente si analizzano tutti i bin dei diagrammi STFT
- si creano matrici (maschere) della stessa dimensione del diagramma STFT
- se il bin che si sta analizzando appartiene alla sorgente in questione, al corrispondente punto della maschera viene assegnato il valore "1"
- in caso contrario al punto viene assegnato il valore "0"



→ La moltiplicazione (applicazione) della maschera di ognuna delle sorgenti al diagramma STFT permette di estrarre dallo stesso solo i punti competenti a quella specifica sorgente; l'applicazione della ISTFT permetta la ricostruzione dei segnali

Performance

→ La valutazione delle PERFORMANCE della procedura implementata è stata effettuata tramite il calcolo del “SIR Improvement”

→ Il SIR è definito come il rapporto tra il segnale utile ed i segnali interferenti

→ Il SIR Improvement è definito come la differenza tra il SIR relativo ad ognuna delle sorgenti prima dell'applicazione dell'algoritmo e del SIR a valle della procedura di separazione

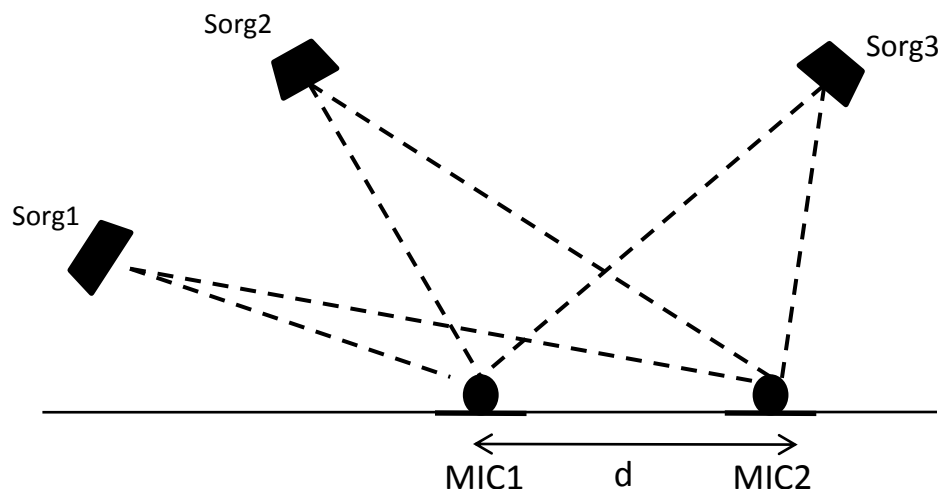
$$SIR_{in}^i = \frac{signal_i}{signal_{i+1} + signal_{i+2}}$$

$$SIR_{out}^i = \frac{signal^{ricostruito}_i}{ISTFT(STFT * masc_i)}$$

$$SIR_improvement = SIR_{out} - SIR_{in}$$

Geometria delle prove

Disposizione Sorgenti/Sensori

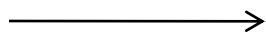


APPROSSIMAZIONI:

- I segnali non vengono attenuati, subiscono solamente ritardo
- Si assume valida l'ipotesi di sparsità delle sorgenti presenti nel segnale captato dai due microfoni; se analizzato nel dominio STFT, si suppone di avere 1 sola sorgente dominante per ogni bin tempo-frequenza

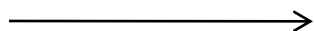
Caratteristiche delle sorgenti e dei ricevitori

Sorg1
Sorg2
Sorg3



Sorgenti di tipo audio

MIC1
MIC2



Microfoni (omnidirezionali)
Freq. di campionamento 8000 kHz

Geometria delle prove

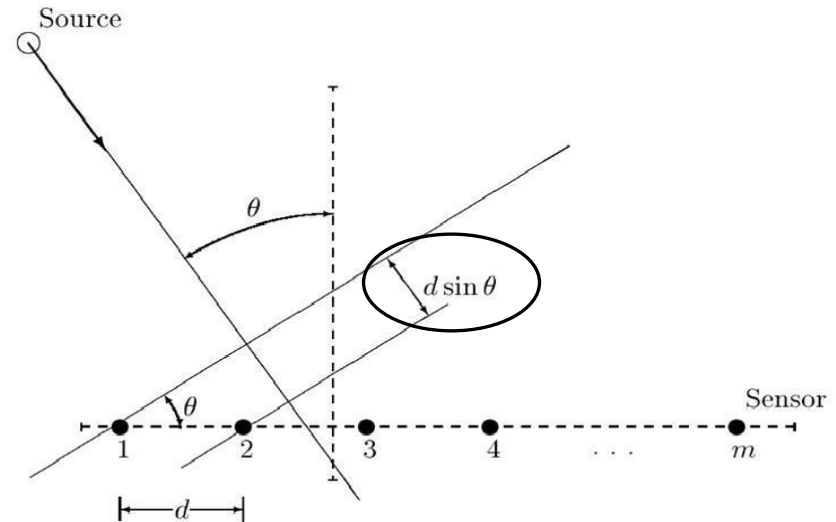
Calcolo del **ritardo**:

$$\tau_i = (k-1) * \frac{d \sin(\theta)}{c}$$

Velocità di propagazione
dell'onda sonora

Nella nostra situazione i sensori sono solo due, per ogni sorgente è necessario il calcolo di un solo ritardo.

Schema geometrico per il calcolo dei ritardi



Calcolo della **distanza tra i sensori**:

→ È stata scelta in base alla minima per evitare alias spaziale

$$d \leq \frac{\lambda}{2} \quad \lambda = \frac{f_{\max}}{c} = \frac{4000 \text{ Hz}}{340 \text{ m/s}} = 8,5 \text{ cm}$$

$$d = 4 \text{ cm}$$

Prima prova (problemi determinati)

Una prima prova è stata effettuata utilizzando due registrazioni sintetiche di due sole sorgenti (due sorgenti per due sensori):

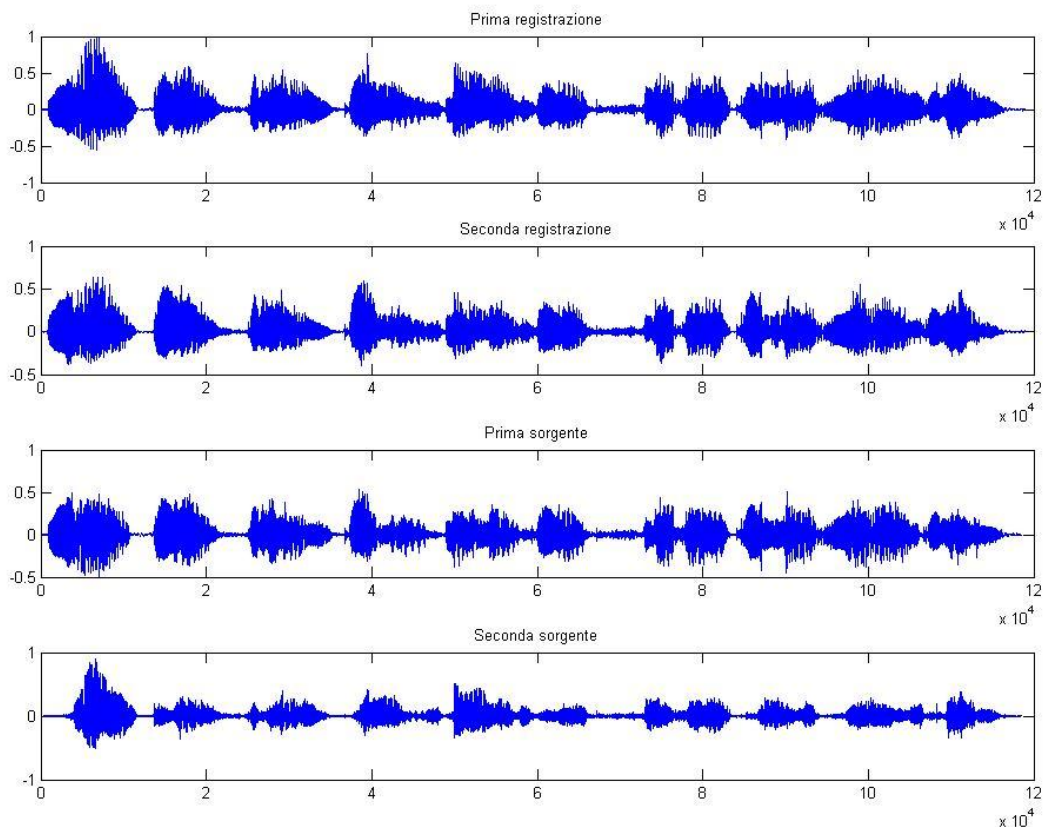
I file utilizzati sono quelli relativi ad un esperimento di separazione cieca di sorgenti tramite l'utilizzo di reti neurali per problemi determinati. Gli stessi file (fonte internet) sono stati utilizzati per verificare il funzionamento dell'algoritmo

[Registrazione1](#)

[Registrazione2](#)

[Separata1](#)

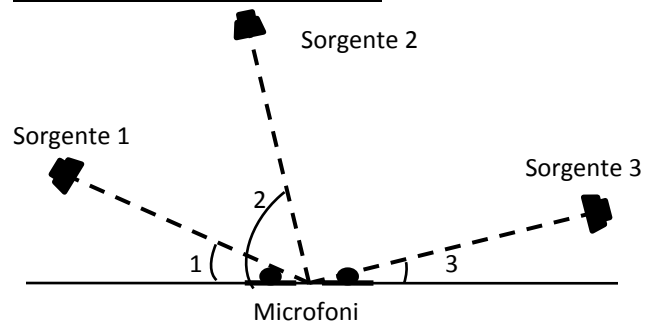
[Separata2](#)



http://cnl.salk.edu/~tewon/Blind/blind_audio.html (il secondo esempio)

Problemi Sottodeterminati

Prima situazione



Angoli (riferimento alla figura):

angolo1 = 30°

angolo2 = 80°

angolo3 = 10°

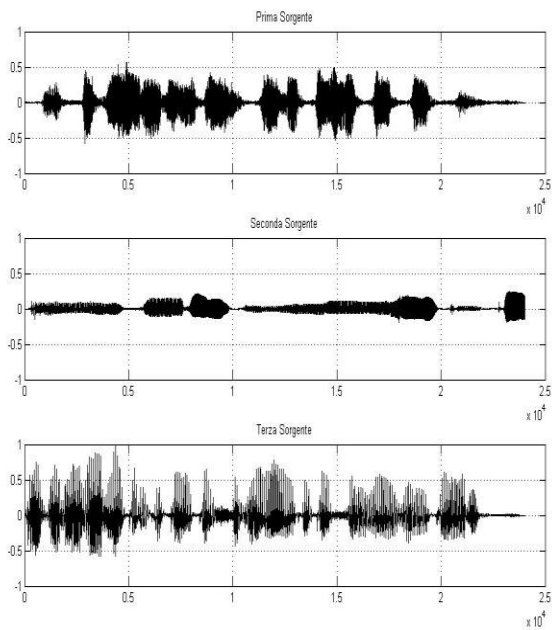
Guadagni in dB (SIR)

G1 = 17.73

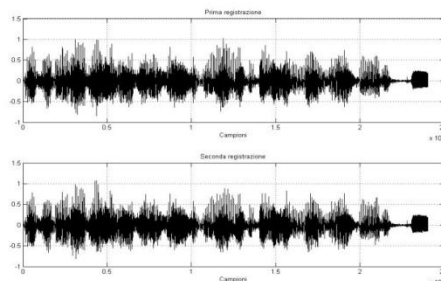
G2 = 14.94

G3 = 15.66

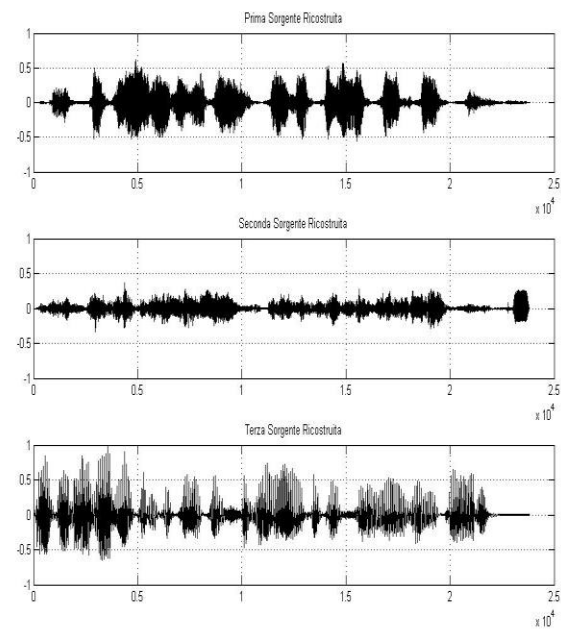
Sorgenti originali



Registrazioni



Sorgenti ricostruite

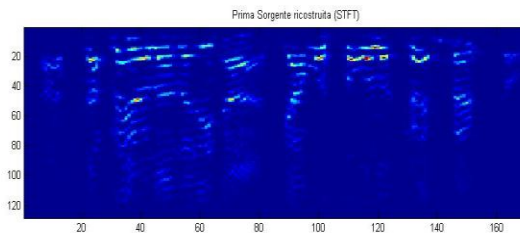
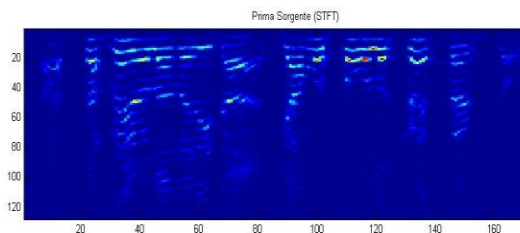


Problemi Sottodeterminati

Prima situazione

Diagrammi STFT

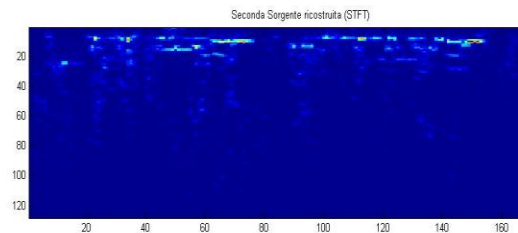
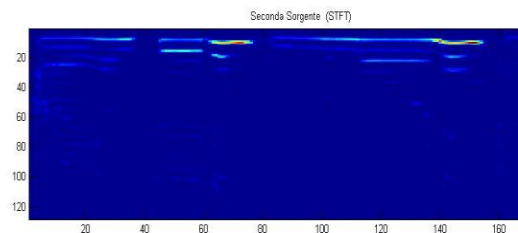
Prima sorgente



[Originale1](#)

[Ricostruita1](#)

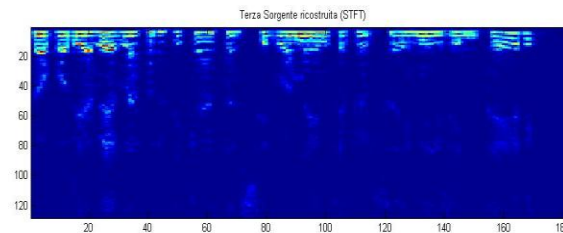
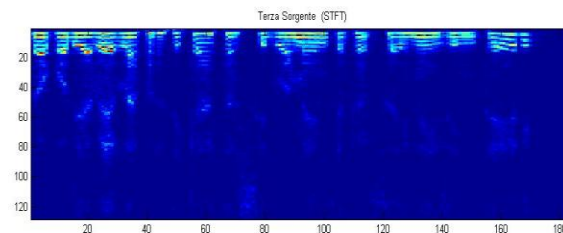
Seconda sorgente



[Originale2](#)

[Ricostruita2](#)

Terza sorgente

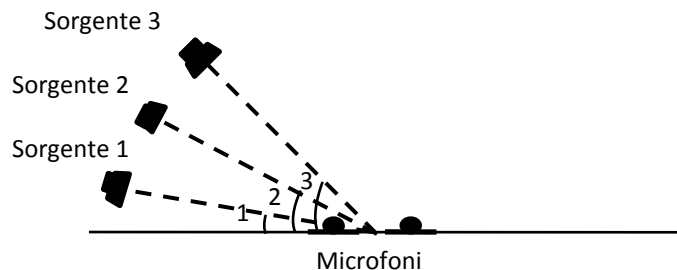


[Originale3](#)

[Ricostruita3](#)

Problemi Sottodeterminati

Seconda situazione



Angoli (riferimento alla figura):

angolo1 = 10°

angolo2 = 20°

angolo3 = 30°

Guadagni in dB (SIR)

G1 = -

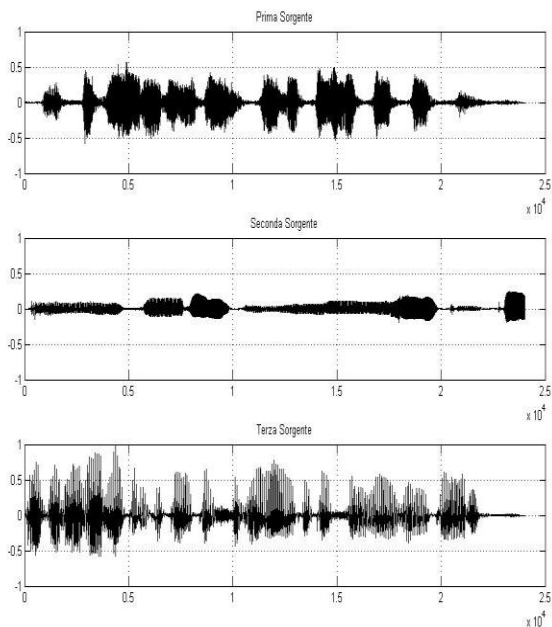
G2 = -

G3 = -

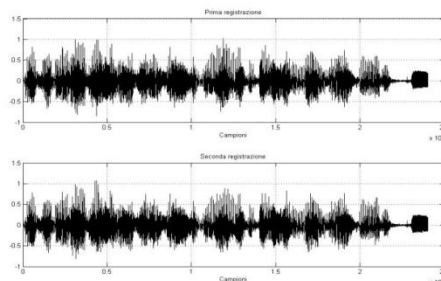
le sorgenti sono molo vicine angularmente

Una delle sorgenti non viene per nulla separata e rimane sovrapposta ad un'altra

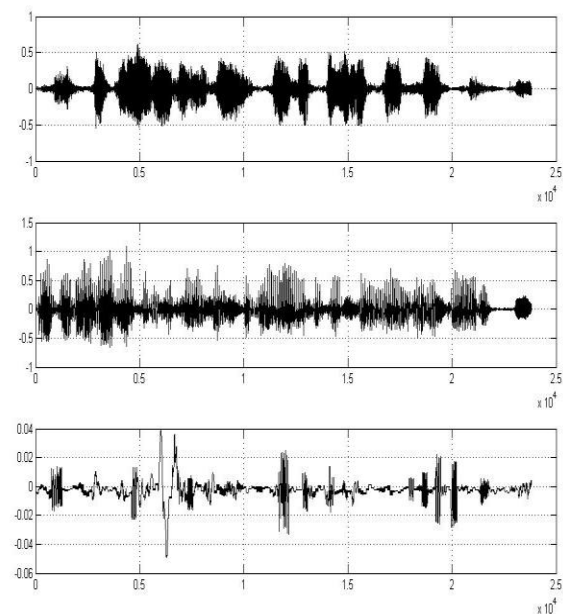
Sorgenti originali



Registrazioni



Sorgenti ricostruite

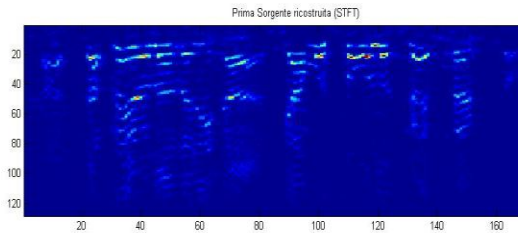
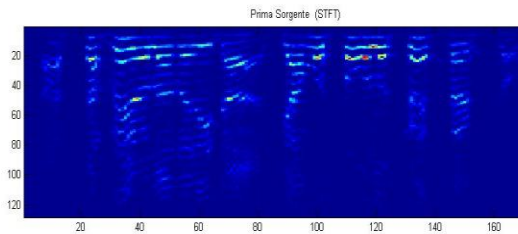


Problemi Sottodeterminati

Seconda situazione

Diagrammi STFT

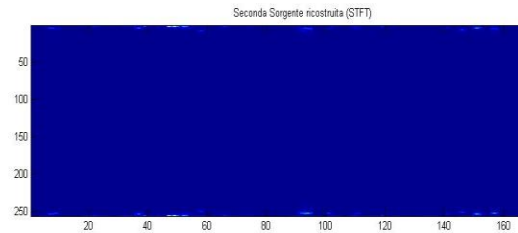
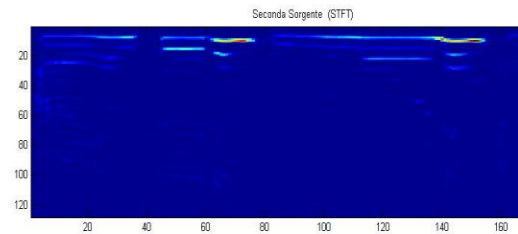
Prima sorgente



[Originale1](#)

[Ricostruita1](#)

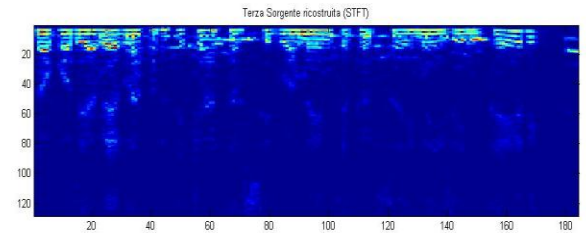
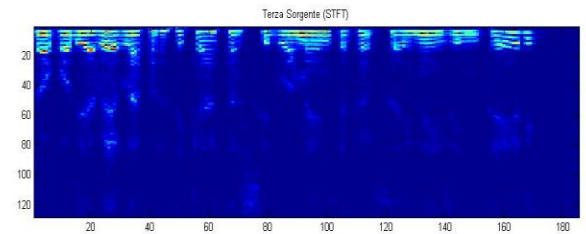
Seconda sorgente



[Originale2](#)

[Ricostruita2](#)

Terza sorgente

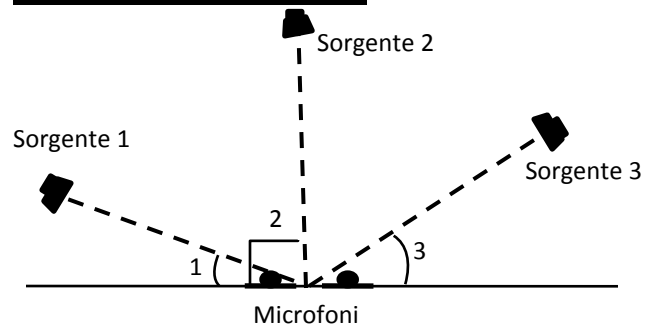


[Originale3](#)

[Ricostruita3](#)

Problemi Sottodeterminati

Terza situazione



Angoli (riferimento alla figura):

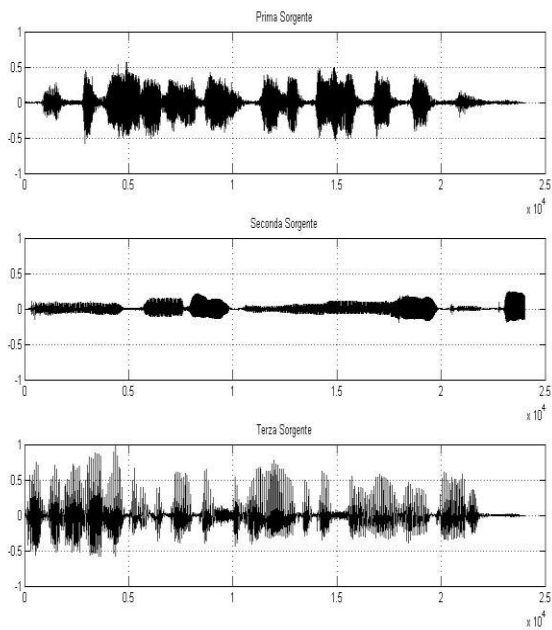
Guadagni in dB (SIR)

G1 = 17.94

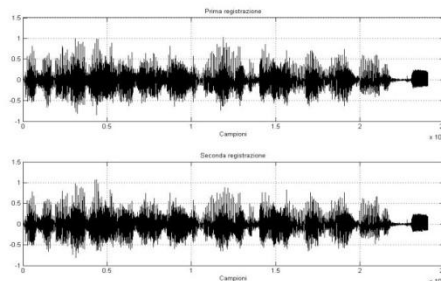
G2 = 14.88

G3 = 15.47

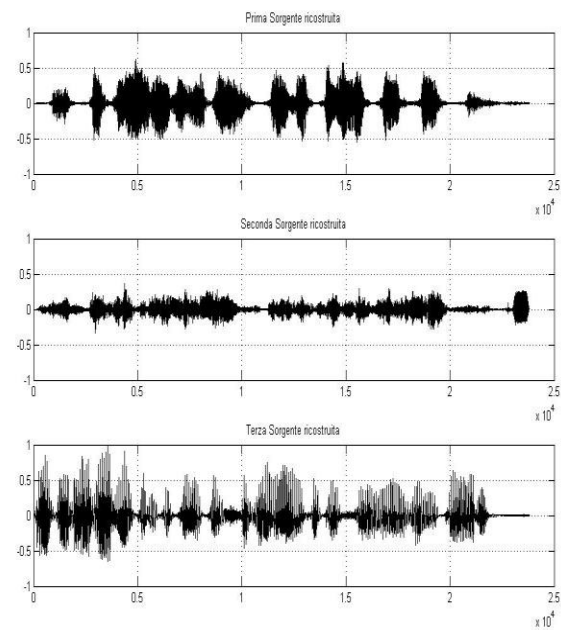
Sorgenti originali



Registrazioni



Sorgenti ricostruite

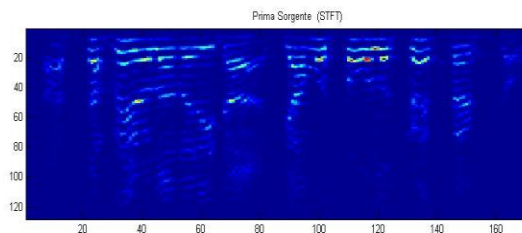


Problemi Sottodeterminati

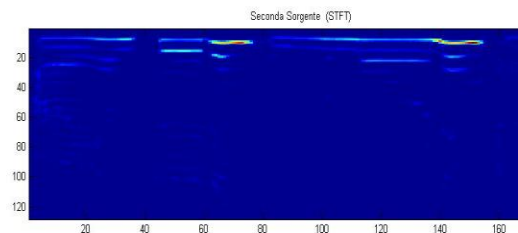
Terza situazione

Diagrammi STFT

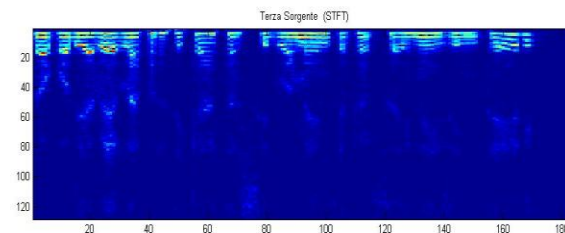
Prima sorgente



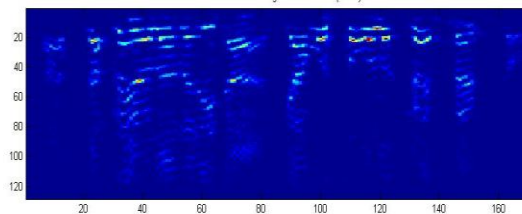
Seconda sorgente



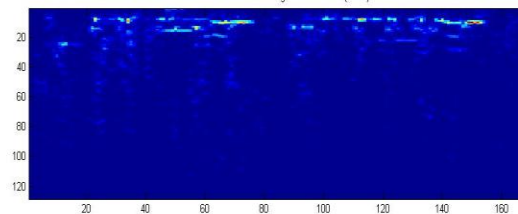
Terza sorgente



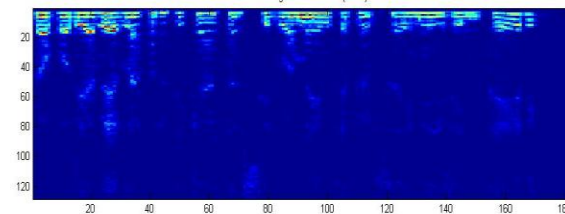
Prima Sorgente ricostruita (STFT)



Seconda Sorgente ricostruita (STFT)



Terza Sorgente ricostruita (STFT)



[Originale1](#)

[Ricostruita1](#)

[Originale2](#)

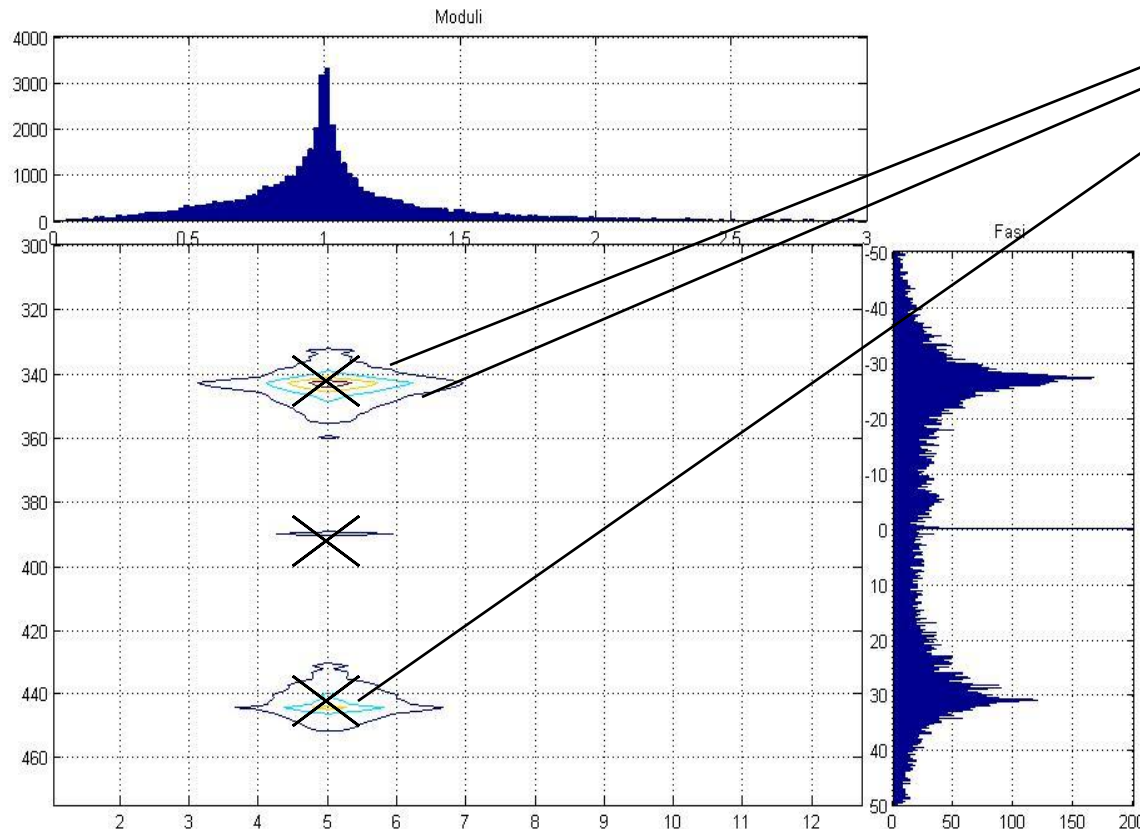
[Ricostruita2](#)

[Originale3](#)

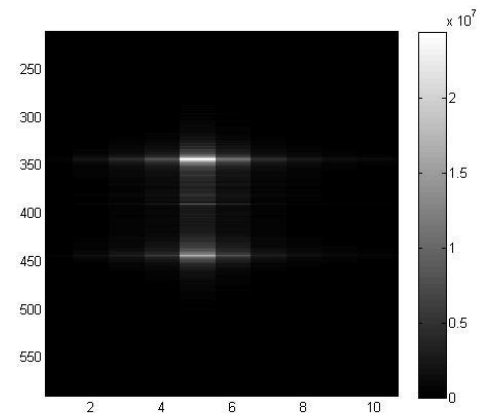
[Ricostruita3](#)

Spazio delle Features

Si riporta un esempio degli istogrammi delle features generati e utilizzati per la classificazione nella prima situazione:



Indicati con "X" sono i punti in cui vengono posizionati i centroidi delle classi create dall'algoritmo di classificazione K-MEANS



I valori sugli assi del diagramma centrale non sono stati normalizzati

Considerazioni Finali

- L'algoritmo implementato funziona e permette di dimostrare l'efficacia della procedura di separazione tramite analisi nel dominio Tempo-Frequenza
- Non sono state effettuate prove in presenza di ambienti riverberanti; ci si aspetta un decremento delle prestazioni per via del venir meno della sparsità delle sorgenti
- L'analisi delle prestazioni ha mostrato come il funzionamento dell'algoritmo dipenda principalmente dalla distanza angolare tra le sorgenti
- La principale limitazione a procedure di questo tipo, basate sull'utilizzo di una classificazione automatica, risiede nel fatto che è necessario conoscere a priori il numero di sorgenti presenti
- Un possibile miglioramento potrebbe essere sviluppato inserendo un algoritmo che, attraverso procedure di DOA, permetta di determinare il numero di sorgenti presenti per poi procedere alla classificazione e separazione