

Confronto fra algoritmo LMS e Fast Deconvolution per la cancellazione del crosstalk

Matteo Orlandini e Jacopo Pagliuca

Università Politecnica delle Marche

28 settembre 2021



Contenuti

- 1 Introduzione
- 2 Stato dell'arte
- 3 Dataset
- 4 LMS
- 5 Fast Deconvolution
- 6 Implementazione NU-Tech
- 7 Risultati

- 1 Introduzione
- 2 Stato dell'arte
- 3 Dataset
- 4 LMS
- 5 Fast Deconvolution
- 6 Implementazione NU-Tech
- 7 Risultati

Cancellazione del crosstalk

Obiettivo

Confrontare gli algoritmi LMS e Fast Deconvolution per la cancellazione del crosstalk.

- Un sistema audio 3D permette di posizionare i suoni intorno ad un ascoltatore in modo che questi siano percepiti come provenienti da punti arbitrari nello spazio.
- Se vengono utilizzati degli altoparlanti, la riproduzione di segnali binaurali all'orecchio dell'ascoltatore non è semplice. Ogni orecchio riceve una componente di diafonia, inoltre, i segnali diretti sono distorti dal riverbero della stanza.
- Per superare i problemi descritti sopra, è necessario un filtro inverso prima di riprodurre il segnale binaurale attraverso gli altoparlanti.

1 Introduzione

2 Stato dell'arte

3 Dataset

4 LMS

5 Fast Deconvolution

6 Implementazione NU-Tech

7 Risultati

- Supponendo che i percorsi di trasferimento acustico dagli altoparlanti alle orecchie siano noti, il metodo di implementazione **diretto** calcola il filtro di cancellazione della diafonia invertendo le funzioni di trasferimento HRTF.
- Nei metodi di implementazione **adattivi**, il filtro di cancellazione della diafonia è calcolato adattando i relativi coefficienti usando i segnali di feedback ricevuti da microfoni in miniatura collocati nelle orecchie dell'utente.
- I metodi diretti o adattativi possono essere implementati nel dominio del **tempo** o della **frequenza**.
- I primi sono generalmente dispendiosi dal punto di vista computazionale, mentre i secondi hanno una complessità inferiore. Gli algoritmi nel dominio del tempo hanno prestazioni migliori di quelli nel dominio della frequenza, data la stessa lunghezza del filtro di cancellazione.

Algoritmi nel dominio del tempo

LMS

Noto per la sua semplicità e robustezza ed è ampiamente utilizzato, anche se la sua velocità di convergenza è lenta.

$$J = E[e[n]^2] = E[(d[n] - y[n])^2]$$

RLS

Si ottiene pesando in modo esponenziale i dati in modo da rimuovere gradualmente gli effetti dei vecchi dati sui coefficienti del filtro e permettere il tracciamento di segnali che variano lentamente.

$$J(\mathbf{h}_n) = \sum_{i=0}^n \lambda^{n-i} e^2[i]$$

Algoritmi nel dominio della frequenza

Fast Deconvolution

La Fast Deconvolution è un metodo molto veloce per calcolare una matrice di filtri digitali che può essere utilizzata per controllare le uscite di un impianto multicanale.

$$J = E + \beta V(f)$$

FLMS

Un algoritmo derivato da LMS è FLMS (Fast Least Mean Square), viene implementato nel dominio della frequenza e richiede uno sforzo computazionale minore rispetto alla sua controparte nel dominio del tempo.

- 1 Introduzione
- 2 Stato dell'arte
- 3 Dataset**
- 4 LMS
- 5 Fast Deconvolution
- 6 Implementazione NU-Tech
- 7 Risultati

Dataset utilizzato

HRTF Measurements of a KEMAR Dummy-Head Microphone

- Dataset del MIT.
- Risposte impulsive dell'orecchio sinistro e destro rispetto ad un altoparlante Realistic Optimus Pro 7 montato a 1,4 metri dal KEMAR.
- I dati HRTF vengono archiviati nelle directory per elevazione. Ogni nome di directory ha il formato "elevEE", dove "EE" è l'angolo di elevazione.
- All'interno di ogni directory, il nome di ogni file ha il formato "XEEeAAAa.wav" dove X può essere "L" o "R" rispettivamente per la risposta dell'orecchio sinistro e destro e "AAA" è l'azimut della sorgente in gradi, da 0° a 355° .

- 1 Introduzione
- 2 Stato dell'arte
- 3 Dataset
- 4 LMS**
- 5 Fast Deconvolution
- 6 Implementazione NU-Tech
- 7 Risultati

In generale, si può scrivere l'uscita $y[n]$ del filtraggio nell' n -esimo istante di un segnale $\mathbf{a}[n]$ definito come:

$$\mathbf{a}[n] = [a[n], a[n-1], \dots, a[n-M+1]]^T$$

con un filtro $\mathbf{b}[n]$ di lunghezza M con coefficienti del tipo:

$$\mathbf{b}[n] = [b_0[n], b_1[n], \dots, b_{M-1}[n]]^T$$

come:

$$y[n] = \sum_{j=0}^{M-1} b_j[n] a[n-j] = \mathbf{b}^T[n] \cdot \mathbf{a}[n]. \quad (1)$$

Il filtraggio dei segnali di riferimento $x_i[n]$ con le hrir c_{lm} è dato dalla seguente equazione:

$$r_{ilm}[n] = \sum_{j=0}^{M-1} c_{lm}[j] x_i[n-j]. \quad (2)$$

Si può applicare la formula (1) all'equazione (2) per calcolare le $r_{ilm}[n]$ nel seguente modo:

$$r_{ilm}[n] = \sum_{j=0}^{M-1} c_{lm,j}[n] x_i[n-j] = \mathbf{c}_{lm}^T[n] \cdot \mathbf{x}_i[n]. \quad (3)$$

Il segnale ricevuto ad ogni orecchio è dato da:

$$y_i[n] = r_{i1}[n] \otimes h_{11}[n] + r_{i2}[n] \otimes h_{21}[n] + r_{i3}[n] \otimes h_{12}[n] + r_{i4}[n] \otimes h_{22}[n]. \quad (4)$$

Il criterio dell'algoritmo LMS è la minimizzazione della funzione costo

$$J = E[e[n]^2] = E[(d[n] - y[n])^2] \quad (5)$$

dove $e[n]$, $d[n]$ e $y[n]$ sono definiti come:

$$\mathbf{e}[n] = \begin{bmatrix} e_1[n] \\ e_2[n] \end{bmatrix}, \quad \mathbf{d}[n] = \begin{bmatrix} d_1[n] \\ d_2[n] \end{bmatrix}, \quad \mathbf{y}[n] = \begin{bmatrix} y_1[n] \\ y_2[n] \end{bmatrix}. \quad (6)$$

La minimizzazione di J avviene con il metodo steepest descend e l'aggiornamento dei tappi del filtro adattativo segue la formula:

$$h^{(k+1)} = h^{(k)} - \mu e_i^{(k)} \cdot \mathbf{r}_i^T. \quad (7)$$

Esplicitando l'equazione (7) per il primo canale, si ottiene:

$$h_{11}^{(k+1)}[n] \leftarrow h_{11}^{(k)}[n] - \mu e_1^{(1)} \cdot r_{111}[n]$$

$$h_{21}^{(k+1)}[n] \leftarrow h_{21}^{(k)}[n] - \mu e_1^{(1)} \cdot r_{112}[n]$$

$$h_{12}^{(k+1)}[n] \leftarrow h_{12}^{(k)}[n] - \mu e_1^{(1)} \cdot r_{211}[n]$$

$$h_{22}^{(k+1)}[n] \leftarrow h_{22}^{(k)}[n] - \mu e_1^{(1)} \cdot r_{212}[n]$$

L'aggiornamento usando il secondo canale avviene in modo analogo.

- 1 Introduzione
- 2 Stato dell'arte
- 3 Dataset
- 4 LMS
- 5 Fast Deconvolution**
- 6 Implementazione NU-Tech
- 7 Risultati

Deconvoluzione

La deconvoluzione, nella sua forma più elementare, può essere descritta come il compito di calcolare l'input di un sistema a tempo discreto conoscendo il suo output.

Consideriamo una funzione costo del tipo:

$$J = E + \beta V(f) \quad (8)$$

dove E è una misura dell'errore della pressione sonora:

$$E = |Y_1 - X_1|^2 + |Y_2 - X_2|^2 \quad (9)$$

e V è una funzione della frequenza che indica il costo computazionale. Il numero $\beta \geq 0$ è un parametro di regolarizzazione che determina quanto peso assegnare alla funzione $V(f)$.

Siano S i segnali trasferiti agli altoparlanti facendo passare il segnale X attraverso la matrice di cancellazione del crosstalk H . Otteniamo:

$$V(f) = S_b^+ S_b \quad (10)$$

con

$$S_b = BS = BHX, \quad (11)$$

dove B è una matrice 2×2 e il simbolo $^+$ indica l'inversa generalizzata della matrice S_b . Se, come nel nostro caso, il dominio della frequenza è discreto e $B = I$, la soluzione approssimata della funzione J è definita da:

$$H[k] = \left[C^H[k]C[k] + \beta I \right]^{-1} C^H[k] \quad (12)$$

dove k indica la k -esima frequenza corrispondente a $\exp(j2\pi k/N)$ e l'apice H denota l'operatore Hermitiano.

Overlap and Save I

Dato che l'uscita Y è data da:

$$Y = CHX = \begin{bmatrix} C_{11}X_1 & C_{12}X_1 & C_{11}X_2 & C_{12}X_2 \\ C_{21}X_1 & C_{22}X_1 & C_{21}X_2 & C_{22}X_2 \end{bmatrix} \begin{bmatrix} H_{11} \\ H_{21} \\ H_{12} \\ H_{22} \end{bmatrix}, \quad (13)$$

nell'implementazione pratica non si può calcolare tutta l'uscita con la sola operazione matriciale (13), occorre usare la tecnica dell'overlap and save per filtrare l'ingresso X con i filtri C e H . L'overlap and save è utile per eseguire un filtraggio in real time con un filtro a risposta impulsiva finita.

Overlap and Save II

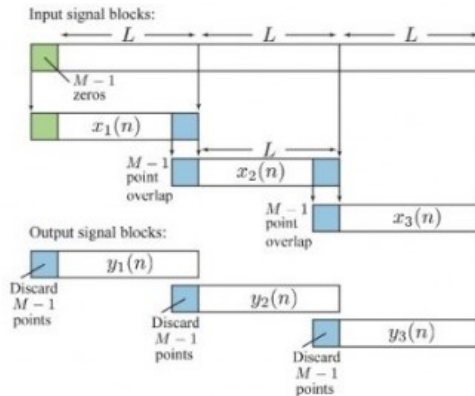


Figura: Metodo Overlap and Save

Overlap and Save III

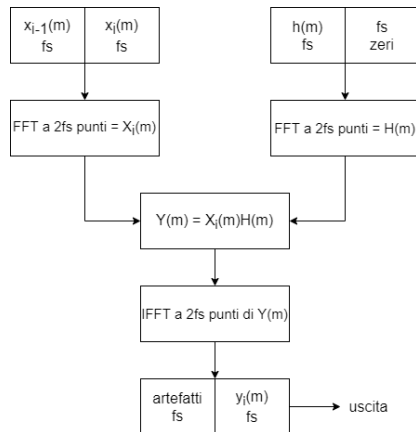


Figura: Diagramma dell'implementazione dell'Overlap and Save

- 1 Introduzione
- 2 Stato dell'arte
- 3 Dataset
- 4 LMS
- 5 Fast Deconvolution
- 6 Implementazione NU-Tech**
- 7 Risultati

LMS

- 1 Inizializzazione delle variabili nel costruttore.
- 2 Allocazione delle variabili e lettura delle HRIR nella Init.
- 3 Costruzione segnale desiderato traslando il segnale di ingresso di τ campioni nella Process.
- 4 Filtraggio dei segnali $x_i[n]$ con le HRIR c_{lm} usando l'equazione (3).
- 5 Calcolo delle uscite con la formula (4).
- 6 Calcolo dell'errore come la differenza fra il segnale desiderato e l'uscita effettiva.
- 7 Aggiornamento dei tappi del filtro di ricostruzione con la formula (7).
- 8 Deallocazione della memoria nella Delete.

Fast Deconvolution

- 1 Inizializzazione delle variabili nel costruttore.
- 2 Allocazione delle variabili e lettura delle HRIR nella Init.
- 3 Calcolo della FFT a fftLen punti delle HRIR.
- 4 Calcolo della matrice \mathbf{H} con la formula (12).
- 5 Costruzione del buffer di ingresso con f_s campioni per l'overlap and save nella Process.
- 6 Calcolo della FFT a fftLen punti del buffer di ingresso.
- 7 Calcolo del buffer di uscita moltiplicando il buffer di ingresso per le HRTF.
- 8 Aggiornamento dell'uscita scartando i primi f_s campioni del buffer.
- 9 Deallocazione della memoria nella Delete.

- 1 Introduzione
- 2 Stato dell'arte
- 3 Dataset
- 4 LMS
- 5 Fast Deconvolution
- 6 Implementazione NU-Tech
- 7 Risultati**

Risultati Matlab

Fattori di separazione dei canali

$$J_L = E \left\{ 20 \cdot \log_{10} \frac{C_{11}H_{11} + C_{12}H_{21}}{C_{21}H_{11} + C_{22}H_{21}} \right\} \quad [\text{dB}] \quad (14)$$

$$J_R = E \left\{ 20 \cdot \log_{10} \frac{C_{22}H_{22} + C_{21}H_{12}}{C_{12}H_{22} + C_{11}H_{12}} \right\} \quad [\text{dB}] \quad (15)$$

μ	J_R [dB]	J_L [dB]
10^{-3}	22.032	21.924
$5 \cdot 10^{-4}$	17.610	20.375
10^{-4}	11.063	16.921
$5 \cdot 10^{-5}$	9.742	16.161

(a) Confronto di J_R e J_L per l'algoritmo LMS per diversi μ .

Azimuth	J_R [dB]	J_L [dB]
$\pm 20^\circ$	21.511	25.532
$\pm 30^\circ$	22.032	21.924
$\pm 40^\circ$	19.667	23.528

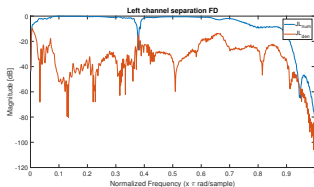
(c) Confronto di J_R e J_L per l'algoritmo LMS per diversi angoli azimutali con $\mu = 10^{-3}$.

β	J_R [dB]	J_L [dB]
1	25.062	26.538
0.3	28.829	30.498
0.1	34.908	36.532
0.01	47.657	48.661

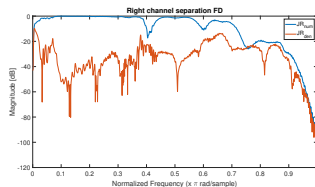
(b) Confronto di J_R e J_L per l'algoritmo FD per diversi β .

Azimuth	J_R [dB]	J_L [dB]
$\pm 20^\circ$	45.672	47.034
$\pm 30^\circ$	34.908	36.532
$\pm 40^\circ$	40.523	42.098

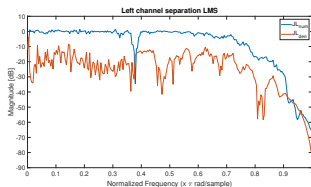
(d) Confronto di J_R e J_L per l'algoritmo FD per diversi angoli azimutali con $\beta = 0.1$.



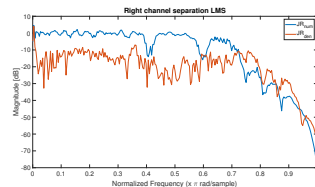
(a)



(b)

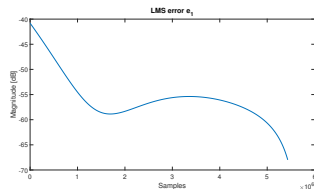


(c)

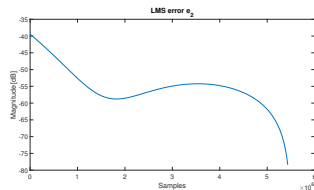


(d)

Figura: Confronto del numeratore e del denominatore di (a) J_L e (b) J_R per FD con $\beta = 0.3$, (c) J_L e (d) J_R per LMS con $\mu = 10^{-3}$.

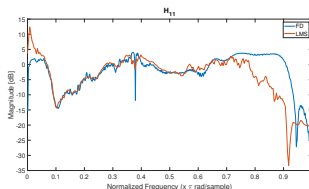


(a)

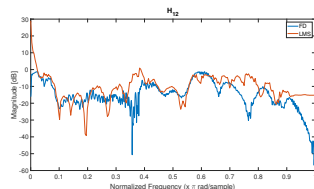


(b)

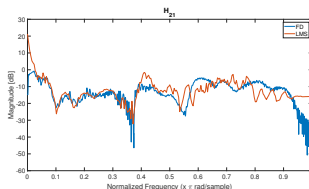
Figura: Confronto dell'MSE del canale (a) sinistro e (b) destro dell'algoritmo LMS.



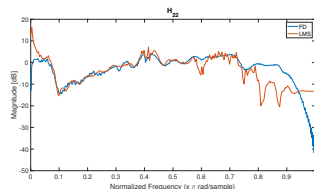
(a)



(b)

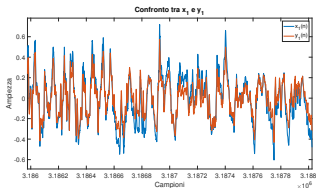


(c)

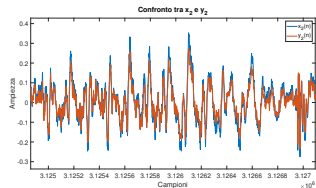


(d)

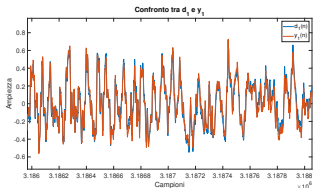
Figura: Confronto dei filtri di cancellazione del crosstalk (a) H_{11} , (b) H_{12} , (c) H_{21} , (d) H_{22} di LMS e Fast Deconvolution.



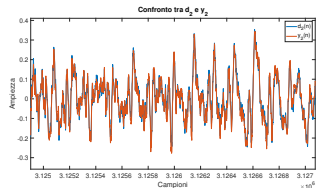
(a)



(b)



(c)



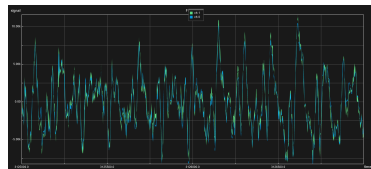
(d)

Figura: Confronto degli ingressi e le uscite (a) sinistro e (b) destro della Fast Deconvolution, (c) sinistro e (d) destro di LMS.

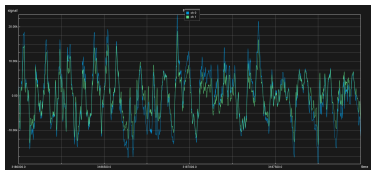
Risultati NU-Tech



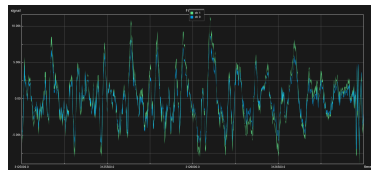
(a)



(b)

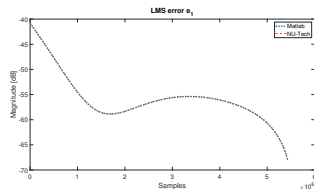


(c)

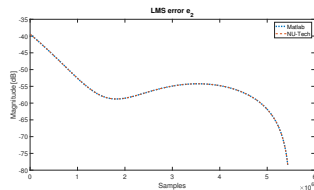


(d)

Figura: Confronto degli ingressi e le uscite (a) sinistro e (b) destro della Fast Deconvolution, (c) sinistro e (d) destro di LMS.

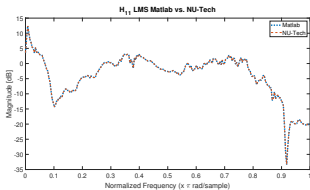


(a)

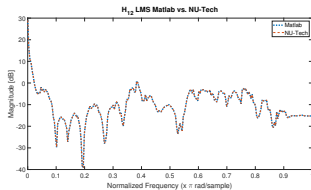


(b)

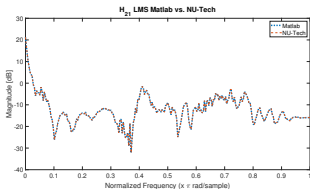
Figura: Confronto dell'MSE del canale (a) sinistro e (b) destro dell'algoritmo LMS.



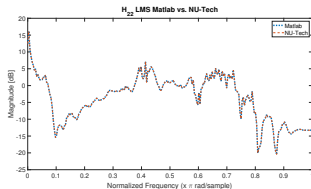
(a)



(b)

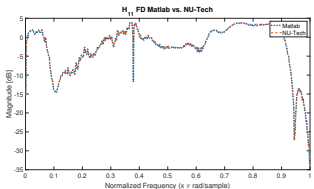


(c)

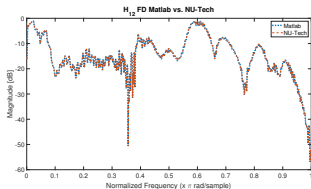


(d)

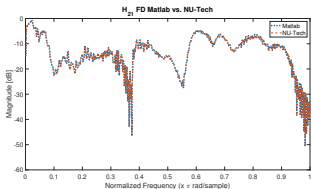
Figura: Confronto dei filtri di cancellazione del crosstalk (a) H_{11} , (b) H_{12} , (c) H_{21} e (d) H_{22} in Matlab e in NU-Tech per l'algoritmo LMS.



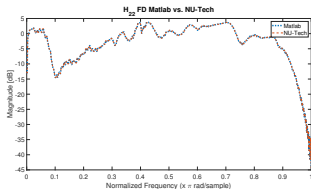
(a)



(b)



(c)



(d)

Figura: Confronto dei filtri di cancellazione del crosstalk (a) H_{11} , (b) H_{12} , (c) H_{21} e (d) H_{22} in Matlab e in NU-Tech per l'algoritmo Fast Deconvolution.

Bibliografia I



Mingsian R. Bai e Chih-Chung Lee. «Objective and subjective analysis of effects of listening angle on crosstalk cancellation in spatial sound reproduction». In: *The Journal of the Acoustical Society of America* 120.4 (2006), pp. 1976–1989. DOI: 10.1121/1.2257986.



E. Ferrara. «Fast implementations of LMS adaptive filters». In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28 (1980), pp. 474–475.



Bill Gardner e Keith Martin. *HRTF Measurements of a KEMAR Dummy-Head Microphone*. Rapp. tecn. 280. (update on October 13, 2006). [Online]. Available: <http://www.media.mit.edu>. MIT Media Lab Perceptual Computing, 1994.

Bibliografia II



A. Gonzalez e J.J. Lopez. «Time domain recursive deconvolution in sound reproduction». In: *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*. Vol. 2. 2000, 11833–11836 vol.2. DOI: 10.1109/ICASSP.2000.859089.



Emmanuel C. Ifeakor e Barrie W. Jervis. *Digital Signal Processing: A Practical Approach*. 2nd. Pearson Education, 2002. ISBN: 0201596199.



Intel®. *Intel® Integrated Performance Primitives*. Dic. 2020.



Jun Seong Kim et al. «A Novel Adaptive Crosstalk Cancellation using Psychoacoustic Model for 3D Audio». In: *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*. Vol. 1. 2007, pp. 1-185-1-188. DOI: 10.1109/ICASSP.2007.366647.



O. Kirkeby et al. «Analysis of ill-conditioning of multi-channel deconvolution problems». In: feb. 1999, pp. 155–158. ISBN: 0-7803-5612-8.

Bibliografia III



O. Kirkeby et al. «Fast deconvolution of multichannel systems using regularization». In: *IEEE Transactions on Speech and Audio Processing* 6.2 (1998), pp. 189–194. DOI: 10.1109/89.661479.



Ole Kirkeby et al. «Design of Cross-Talk Cancellation Networks by Using Fast Deconvolution». In: (nov. 2000).



Chris Kyriakakis. «Fundamental and Technological Limitations of Immersive Audio Systems». In: *Proceedings of the IEEE* 86 (giu. 1998), pp. 941–951. DOI: 10.1109/5.664281.



Dan Li et al. «Comprehensive comparison of the least mean square algorithm and the fast deconvolution algorithm for crosstalk cancellation». In: *2012 International Conference on Audio, Language and Image Processing*. 2012, pp. 224–229. DOI: 10.1109/ICALIP.2012.6376616.

Bibliografia IV



A. Mouchtaris et al. «Inverse filter design for immersive audio rendering over loudspeakers». In: *IEEE Transactions on Multimedia* 2.2 (2000), pp. 77–87. DOI: 10.1109/6046.845012.



P.A. Nelson et al. «Adaptive inverse filters for stereophonic sound reproduction». In: *IEEE Transactions on Signal Processing* 40.7 (1992), pp. 1621–1632. DOI: 10.1109/78.143434.



Lin Wang e Yin Fuliang. «A Stereo Crosstalk Cancellation System Based on the Common-Acoustical Pole/Zero Model». In: *EURASIP Journal on Advances in Signal Processing* 2010 (gen. 2010). DOI: 10.1155/2010/719197.