



UNIVERSITA' DEGLI STUDI DI PADOVA
FACOLTA' DI INGEGNERIA

CORSO DI LAUREA TRIENNALE IN INGEGNERIA MECCATRONICA

ALGORITMI ADATTATIVI PER LA CANCELLAZIONE DEL RUMORE

RELATORE: CH.MO PROF. ALESSANDRO CHIUSO

LAUREANDO: MARCO DAL ZOTTO

ANNO ACCADEMICO: 2010/2011

ALGORITMI ADATTATIVI PER LA CANCELLAZIONE DEL RUMORE

Marco Dal Zotto

Ai miei genitori

*L'uomo stolto cerca la felicità
lontano, il saggio la fa crescere
sotto i propri piedi.*

ROBERT OPPENHEIMER

Indice

Introduzione	1
1. Algoritmi adattativi	3
1.1. Il metodo del gradiente	4
1.2. Algoritmo LMS	6
1.2.1. Struttura dell'algoritmo	7
1.2.2. La scelta del parametro μ	7
1.3. Varianti dell'algoritmo LMS	9
1.3.1. Algoritmo LMS Normalizzato	9
1.3.2. Algoritmo <i>Leaky LMS</i>	12
1.4. Ambiti applicativi	12
1.4.1. Cancellazione adattativa del rumore	13
1.4.2. Identificazione adattativa del sistema	14
2. Controllo attivo feedforward del rumore a banda larga	17
2.1. Effetto del percorso secondario	20
2.2. Algoritmo FXLMS	22
2.2.1. Derivazione dell'algoritmo	23
2.2.1.1. Tecnica di modellizzazione off-line	26
2.3. Simulazioni in MATLAB di FXLMS	28
2.3.1. Primo caso: conoscenza esatta di $S(z)$	29
2.3.2. Secondo caso: stima di $S(z)$	31
3. ANC - Applicazione ai caschi per motoveicoli	33
3.1. Natura del rumore percepito	33
3.2. Analisi del rumore	34
3.2.1. Posizionamento dei sensori	36
3.3. Applicazione dell'algoritmo FXLMS	38

3.3.1. Primo caso	38
3.3.1.1. Risultato della simulazione	39
3.3.2. Secondo caso	40
3.3.2.1. Risultato della simulazione	40
3.3.3. Terzo caso	41
3.3.3.1. Modellizzazione di $S(z)$	42
3.3.3.2. Risultato della simulazione	43
Conclusioni	45
A. Sorgenti in linguaggio Matlab	47
A.1. Algoritmo LMS	47
A.2. Algoritmo NLMS	47
A.3. Algoritmo Leaky LMS	48
A.4. Algoritmo FXLMS	49
B. Simbologia e nomenclatura	51
Bibliografia	53

Introduzione

L'esposizione al rumore acustico è una delle principali cause di malattie professionali e di disagio nel mondo del lavoro ma anche nella quotidianità. In ambito professionale normative introdotte negli ultimi anni hanno imposto un uso più massivo di dispositivi di protezione individuale anche per limitare la percezione del rumore dovuto al funzionamento di macchinari di vario genere. Anche negli uffici la sempre più numerosa presenza di impianti di climatizzazione e condizionamento dell'aria porta irrimediabilmente ad un aumento dell'esposizione delle persone al rumore acustico.

La tendenza odierna a costruire edifici con materiali sempre più leggeri e sottili ha portato la popolazione ad essere sempre più esposta a rumori. Anche nei mezzi di trasporto, la sensazione di comfort dipende sia dalla quantità di rumore percepita dai passeggeri sia dalle vibrazioni meccaniche. Tali vibrazioni sono un tipo di rumore che comunemente crea delle problematiche sia nei mezzi di trasporto che nei macchinari industriali.

Inoltre non bisogna dimenticare la sempre più popolosa presenza di elettrodomestici i quali se da un lato hanno semplificato, velocizzato e migliorato le quotidiane faccende, dall'altro hanno irrimediabilmente contribuito ad aggravare l'inquinamento acustico in ambiente domestico.

Il rumore può essere classificato in base alla dispersione energetica in frequenza:

Rumore a banda larga: detto anche *Broadband noise*, distribuisce l'energia più o meno uniformemente nell'orizzonte frequenziale. È totalmente casuale ed è causato generalmente da fenomeni turbolenti. Esempi di questo tipo di rumore sono quelli a bassa frequenza degli aerei e quelli impulsivi delle esplosioni.

Rumore a banda stretta: detto anche *Narrowband noise*. Questa tipologia di rumore concentra la maggior parte dell'energia intorno ad una specifica

frequenza. Questi rumori sono in genere causati dalle macchine rotanti o ripetitive come ad esempio turbine, compressori e refrigeratori.

L'approccio tradizionale al controllo del rumore acustico si basa sull'utilizzo di tecniche passive quali ad esempio barriere acustiche. I silenziatori passivi effettuano una buona attenuazione in un ampio spettro di frequenza, tuttavia risultano essere grandi, costosi ed inefficaci alle basse frequenze in quanto la lunghezza d'onda del segnale acustico risulta essere grande rispetto alle dimensioni del silenziatore stesso. Essi si basano su due principi di funzionamento e per questo possono essere divisi in due grossi rami:

Silenziatori resistivi: sfruttano la perdita di energia causata dalla propagazione dell'onda acustica entro condotti foderati con materiali fono-assorbenti.

Silenziatori reattivi: sfruttano il concetto di cambiamento di impedenza dovuto alla combinazione di deflettori e tubi per silenziare il rumore indesiderato.

Un'alternativa alle tecniche passive è rappresentata dal controllo attivo del rumore o *Active Noise Control* (ANC). Questa tecnica si basa sul principio di sovrapposizione di due segnali tra loro opposti. In pratica dato un rumore indesiderato con una certa forma d'onda (rumore primario) un certo sistema costruisce attraverso strumenti opportuni un segnale di rumore secondario (detto *antinoise*) che sovrapposto al primo tenta di annullarlo. Questa tecnica risulta essere molto valida laddove le tecniche di riduzione del rumore di tipo passivo risultano inefficaci (cioè alle basse frequenze). D'altro canto le prestazioni delle tecniche attive alle alte frequenze sono limitate da diversi fattori, inclusa la necessità di dover campionare il segnale in ingresso con velocità di campionamento elevate.

Lo scopo di questa tesi è quello di introdurre e analizzare la tecnica ANC. Essa verrà valutata dal punto di vista dell'elaborazione dei segnali più che dal punto di vista acustico. Nel primo capitolo si farà una panoramica dei principali algoritmi adattativi utilizzati per la realizzazione dei filtri adattativi digitali, si proseguirà poi con l'introduzione del controllo attivo feedforward del rumore a banda larga che riassume in sé molti problemi di interesse pratico e si concluderà con un capitolo riguardante l'applicazione del controllo attivo del rumore ai caschi per motoveicoli.

Capitolo 1.

Algoritmi adattativi

Nel contesto dell'elaborazione dei segnali, con il termine *filtraggio lineare* si intende un processo lineare atto all'alterazione del contenuto spettrale di un segnale in un modo ben preciso. Lo strumento che permette di eseguire questa alterazione viene detto *filtro*. Quando il filtro si trova ad operare su campioni di segnale discreti nel tempo si parla di *filtro digitale* (o filtro numerico), il quale permette di effettuare operazioni anche molto complesse.

Quando i parametri propri del filtro, come per esempio la larghezza di banda, non rimangono costanti nel tempo ma vengono fatti variare in maniera tale da adattarsi alle variazioni si parla di *filtri adattativi* (figura 1.1) e il sistema che si occupa di supervisionare questa variazione, che viene attuata per mezzo di un continuo aggiornamento dei coefficienti del filtro, si chiama *algoritmo adattativo*. È facile quindi intuire che nei filtri adattativi, a differenza dei filtri convenzionali, l'ipotesi di tempo-invarianza viene a mancare.

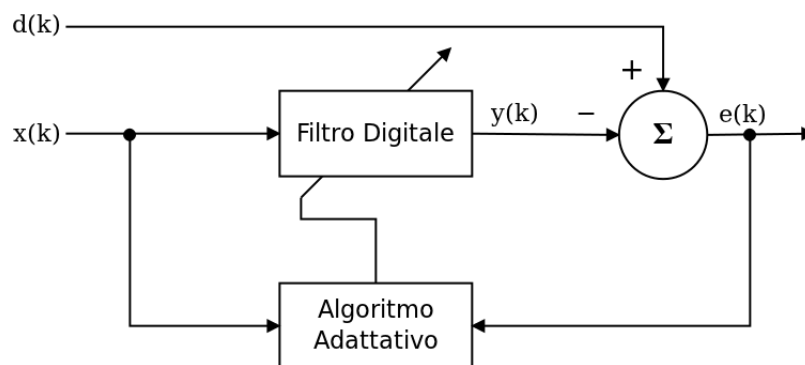


Figura 1.1.: Schema a blocchi di un filtro adattativo.

Dalla figura 1.1 si può vedere che l'algoritmo adattativo riceve in ingresso lo

stesso segnale $x(k)$ ¹, detto segnale primario, che viene posto in ingresso al filtro digitale e un segnale $e(k)$, detto segnale di errore, risultante dalla differenza tra il segnale $d(k)$, detto segnale di riferimento, e il segnale $y(k)$ in uscita dal filtro digitale. Si può intuire che $y(k)$ dipenderà strettamente sia dal segnale $x(k)$ sia dal segnale $e(k)$ (e quindi da $d(k)$).

1.1. Il metodo del gradiente

Il metodo del gradiente viene anche detto *method of steepest descent*. L'obiettivo di questo metodo è quello di cercare di rendere nullo il segnale d'errore, ovvero far sì che $e(k) \cong 0$. Questo si ottiene tramite l'aggiornamento dei coefficienti del filtro digitale. La misura delle prestazioni di tale metodo viene fatta basandosi sulla stima dell'errore quadratico medio o MSE (*mean-square error*) definito come:

$$J(k) := E [e^2(k)] \quad (1.1)$$

dove l'operando $E[\cdot]$ si riferisce al valore atteso.

Si può dimostrare che, per un filtro FIR², MSE è una funzione quadratica dei pesi che può essere rappresentata come una superficie iperbolica convessa. L'obiettivo degli algoritmi adattativi è quello di aggiustare i coefficienti (o pesi) del filtro in maniera tale da minimizzare l'errore, ovvero percorrere la superficie verso il basso fino a raggiungere un punto di minimo nel quale il vettore dei pesi avrà il suo ottimo. Per fare ciò, generalmente, gli algoritmi eseguono una stima del gradiente e si muovono in modo incrementale verso il basso.

Esistono vari algoritmi che eseguono questo tipo di operazione (come per esempio il metodo di *Newton*, il metodo del gradiente,...). La scelta di un algoritmo rispetto ad un altro viene fatta principalmente in base a due fattori discriminanti:

1. la *velocità di convergenza*
2. la *complessità computazionale*

¹Per le varie convenzioni sulla simbologia e la nomenclatura utilizzate nella presente tesi si veda l'appendice B.

²Il termine FIR (*Finite Impulse Response*) sta ad indicare una tipologia di filtri digitali che se sollecitati in ingresso da un impulso unitario discreto (o impulso di Kronecker) producono in uscita una risposta finita.

L'impulso di Kronecker è definito come $\delta(k) := \begin{cases} 1, & k = 0 \\ 0 & k \in \mathbb{Z} \setminus \{0\} \end{cases}$

Il metodo di Newton consente di calcolare molto rapidamente uno zero di una funzione assegnata, in questo caso $e(\cdot)$, cioè un punto \bar{k} tale che la funzione abbia valore nullo in quel punto.

Affinché il metodo sia applicabile è necessario che esista un intervallo $[a, b]$ tale che:

1. il punto \bar{k} appartenga a tale intervallo, ovvero $\bar{k} \in [a, b]$.
2. le derivate prima e seconda della funzione siano non nulle per tutti i punti dell'intervallo, cioè $e'(k) \neq 0$, $e''(k) \neq 0 \forall k \in [a, b]$.

Per utilizzare praticamente il metodo di Newton, il primo problema è quindi individuare un intervallo $[a, b]$ in cui lo zero di $e(k)$ sia contenuto con certezza e in cui le derivate prima e seconda di $e(k)$ non si annullino. Questo problema va risolto caso per caso studiando la funzione in esame e costituisce il principale limite del metodo: per questo motivo tale metodo non verrà utilizzato negli algoritmi trattati nella presente tesi.

Il metodo del gradiente è, invece, una tecnica iterativa che viene utilizzata nella programmazione non lineare e nei problemi di ottimizzazione per trovare una soluzione che minimizzi una funzione obiettivo. La tecnica basa il suo funzionamento su correzioni successive del vettore dei pesi nella direzione del gradiente massimo della superficie convessa. Ad un certo punto si arriverà al suo punto di minimo J_{min} nel quale il vettore dei pesi assumerà il suo valore ottimale.

Sia $J(0)$ il valore di MSE all'istante $k = 0$ e sia $\mathbf{w}(0)$ il valore iniziale, arbitrario, del vettore dei pesi \mathbf{w} . L'idea del metodo è quella di spostarsi nella superficie di errore in direzione del gradiente nel punto. I pesi del filtro sono aggiornati ad ogni iterazione nella direzione del gradiente negativo della superficie di errore.

Matematicamente l'algoritmo può essere espresso come:

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \frac{\mu}{2} \nabla \mathbf{J}(k) \quad (1.2)$$

dove il parametro μ , che viene detto passo o *step-size*, controlla la stabilità e la velocità di discesa verso il punto di minimo³. Il vettore $\nabla J(k)$ indica il gradiente della funzione di errore rispetto a $\mathbf{w}(k)$ e all'incremento di segno negativo del vettore dei pesi in direzione del gradiente negativo.

³Il fattore (arbitrario) 2 verrà incluso ai fini di semplificare la forma finale dell'algoritmo.

Di seguito verranno introdotti i vari algoritmi adattativi partendo dall'algoritmo LMS (*least-mean-square algorithm*) da cui saranno poi derivati ulteriori algoritmi modificati al fine di ottenere un miglioramento delle prestazioni.

1.2. Algoritmo LMS

Dall'equazione (1.2) si può vedere che l'incremento da $\mathbf{w}(k)$ a $\mathbf{w}(k+1)$ è nella direzione del gradiente negativo, quindi il vettore dei pesi seguirà approssimativamente il percorso di discesa più rapida nella superficie delle performance.

In molte applicazioni pratiche le statistiche dei segnali $x(k)$ e $d(k)$ sono sconosciute e quindi non è possibile utilizzare direttamente il metodo del gradiente. Infatti non è possibile calcolare direttamente il valore atteso del segnale di errore. Per ovviare a ciò si cerca di *stimare* l'errore quadratico medio $J(k)$. Una possibile soluzione è quella di utilizzare l'errore quadratico istantaneo, $e^2(k)$, per stimare l'errore quadratico medio definito nella (1.1).

$$\hat{J}(k) = e^2(k) \quad (1.3)$$

Pertanto il gradiente stimato usato nell'algoritmo LMS è semplicemente il gradiente istantaneo dell'errore quadratico del singolo campione:

$$\nabla \hat{J}(k) = 2 [\nabla e(k)] e(k) \quad (1.4)$$

Dato che $e(k) = d(k) - \mathbf{w}^T(k) \mathbf{x}(k)$, con $\mathbf{x}(k) = [x(k), x(k-1), \dots, x(k-L+1)]^T$, $\mathbf{w}(k) = [w_1(k), w_2(k), \dots, w_L(k)]$, L ordine del filtro, si ottiene:

$$\nabla e(k) = -\mathbf{x}(k)$$

e la stima del gradiente risulta quindi essere pari a:

$$\nabla \hat{J}(k) = -2\mathbf{x}(k) e(k) \quad (1.5)$$

Sostituendo la stima del gradiente dell'equazione (1.5) nell'algoritmo del gradiente espresso dall'equazione (1.2) si ottiene:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu \mathbf{x}(k) e(k) \quad (1.6)$$

L'equazione (1.6) rappresenta l'algoritmo LMS. Questo algoritmo è semplice e non richiede operazioni di elevamento al quadrato, media o differenziazione che sono dispendiose da eseguire su un calcolatore.

1.2.1. Struttura dell'algoritmo

Per il funzionamento dell'algoritmo è necessario scegliere a priori i parametri e le condizioni iniziali ovvero L , μ e $\mathbf{w}(0)$ dove L rappresenta l'ordine del filtro, μ il passo e $\mathbf{w}(0)$ il vettore iniziale dei pesi al tempo $k = 0$.

In secondo luogo si deve calcolare l'uscita del filtro adattativo come segue:

$$y(k) = \sum_{i=0}^{L-1} w_i(k) x(k-i)$$

e calcolare il segnale di errore $e(k)$:

$$e(k) = d(k) - y(k)$$

Alla fine si possono aggiornare i coefficienti del vettore dei pesi da $\mathbf{w}(k)$ a $\mathbf{w}(k+1)$ usando l'algoritmo LMS:

$$w_i(k+1) = w_i(k) + \mu x(k-i) e(k), \quad i = 0, 1, \dots, L-1$$

Il codice sorgente in linguaggio MATLAB del filtro digitale che utilizza l'algoritmo LMS per l'aggiornamento dei pesi è presentato in Appendice A.1.

1.2.2. La scelta del parametro μ

Nell'utilizzo dell'algoritmo LMS si deve prestare particolare attenzione alla scelta dello step-size μ . Si può dimostrare [1] che, ai fini della convergenza della media dei pesi, il parametro μ deve rientrare nelle seguenti specifiche:

$$0 < \mu < \frac{2}{LP_x} \tag{1.7}$$

dove P_x indica la potenza del segnale di ingresso $x(k)$.

La convergenza della varianza dei pesi, ovvero dell'errore quadratico medio (o *Mean Square Error, MSE*) impone una condizione più rigorosa. Per un segnale

gaussiano⁴ la convergenza dell'errore quadratico medio richiede che μ soddisfi alla condizione:

$$0 < \mu < \frac{2}{3LP_x} \quad (1.8)$$

Le equazioni precedenti forniscono alcune informazioni fondamentali per la scelta del fattore di convergenza μ .

1. Dato che il limite superiore di μ è inversamente proporzionale a L , un parametro μ piccolo viene utilizzato per filtri con un ordine L elevato.
2. Dato che μ è inversamente proporzionale alla potenza del segnale di ingresso, per segnali con potenza elevata deve essere utilizzato un fattore di convergenza piccolo mentre per segnali con potenza non elevata può essere usato un fattore di convergenza grande.

Da quanto emerso è convenzione utilizzare un fattore di convergenza μ che soddisfi alla seguente condizione [1]:

$$\frac{0.01}{LP_x} < \mu < \frac{0.1}{LP_x}$$

Nelle seguenti simulazioni si vuole mostrare come la scelta del parametro μ influenzi il tempo di convergenza. Per la nomenclatura dei vari parametri ci si riferisca alla figura 1.1.

I dati utilizzati per le simulazioni sono:

Segnale di riferimento: $d(k) = \sin(2\pi \cdot k)$

Segnale in ingresso: $x(k) = \sin(2\pi \cdot k) + 0.1\cos(2\pi \cdot 50 \cdot k)$

Ordine del filtro: $L = 5$

Le due simulazioni differiscono per la scelta del fattore di convergenza μ : per la prima è stato scelto un fattore di convergenza pari a $\mu_{MAX} = \frac{0.1}{L \cdot P_x}$, per la seconda invece è stato scelto un parametro $\mu = 0.5$ che, in questo caso, porta alla divergenza del vettore dei pesi.

I grafici relativi alle simulazioni sono riportati di seguito. La figura 1.2a rappresenta il primo caso, cioè $\mu = \mu_{MAX}$. In questo caso il vettore dei pesi converge

⁴Per segnale gaussiano si intende un segnale aleatorio che ha come funzione di densità di probabilità una distribuzione normale (o Gaussiana).

verso i valori corretti. Nel secondo caso, illustrato in figura 1.2b, ciò non succede. Si vede infatti che i pesi del vettore oscillano senza giungere a dei valori determinati. Per una migliore comprensione si è scelto di mostrare solamente i primi $k = 100$ campioni.

1.3. Varianti dell'algoritmo LMS

L'algoritmo LMS descritto nella precedente sezione è il più usato nelle applicazioni pratiche. In questa sezione si presenteranno una serie di algoritmi derivanti dall'LMS basilare. Questi algoritmi sono da ritenersi delle varianti in quanto sono frutto di studi per migliorare alcune caratteristiche intrinseche dell'algoritmo LMS quali la velocità di convergenza, la semplicità di implementazione, la robustezza di funzionamento.

1.3.1. Algoritmo LMS Normalizzato

La stabilità, il tempo di convergenza e le fluttuazioni dell'algoritmo LMS sono governate dal parametro μ , che esprime il fattore di convergenza, e quindi, come emerso dalle equazioni (1.7) e (1.8), dalla *potenza* del segnale di riferimento.

Una tecnica per ottimizzare la velocità di convergenza indipendentemente dalla potenza del segnale di riferimento di ingresso $x(k)$, è conosciuta come Algoritmo LMS Normalizzato (o *Normalized LMS algorithm*, NLMS) il quale è espresso dall'equazione:

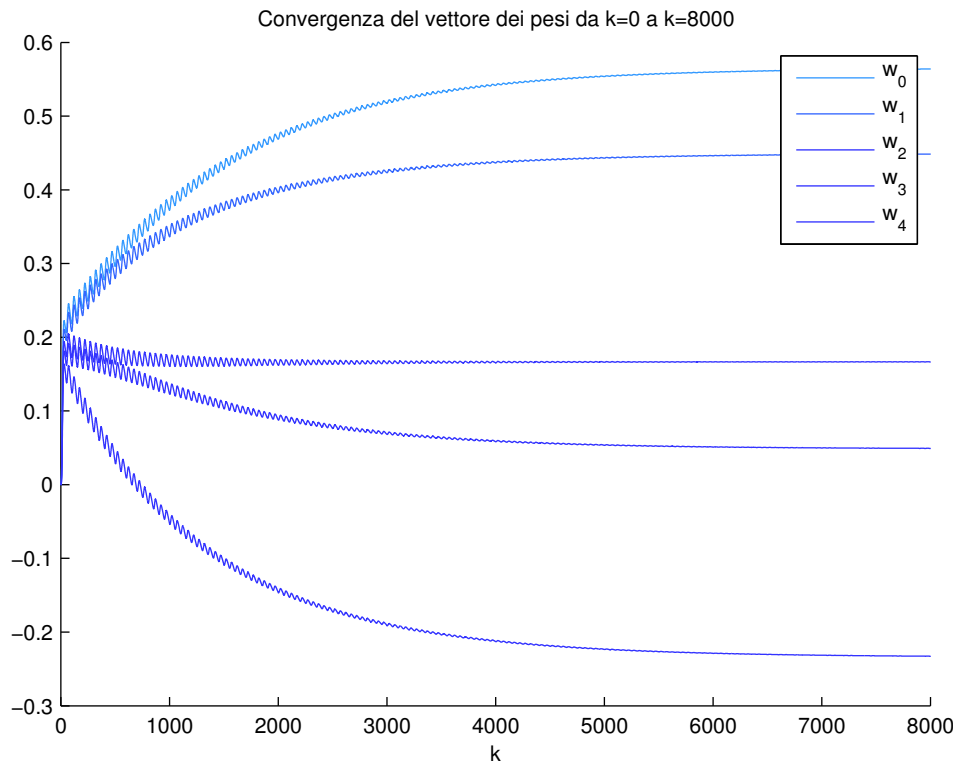
$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu(k) \mathbf{x}(k) e(k) \quad (1.9)$$

dove $\mu(k)$ è il *passo adattativo* che viene calcolato come:

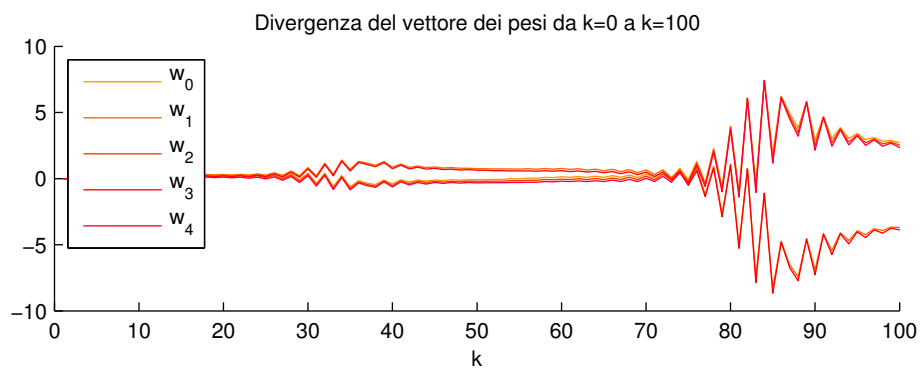
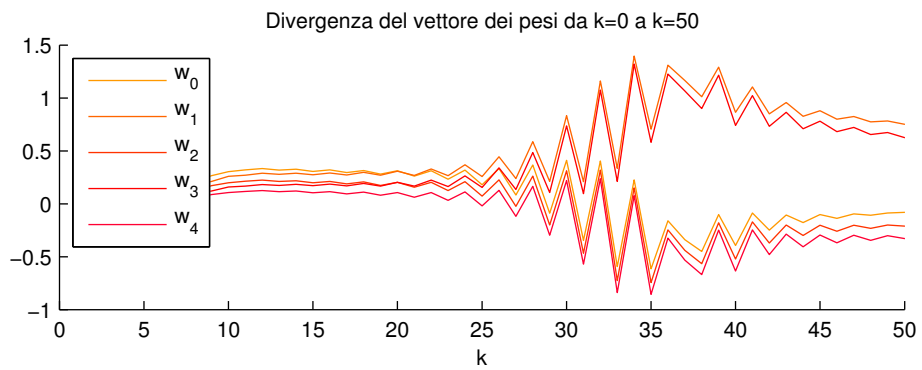
$$\mu(k) = \frac{\alpha}{L\hat{P}_x(k)} \quad (1.10)$$

dove $\hat{P}_x(k)$ è la stima della potenza del segnale $x(k)$ al tempo k , e α è il *passo normalizzato* il quale rientra nell'intervallo:

$$0 < \alpha < 2$$



(a) Convergenza dei pesi: $\mu = \mu_{MAX}$



(b) Divergenza del vettore dei pesi: $\mu = 0.5$.

Figura 1.2.: Variazione dei coefficienti del vettore dei pesi in base al valore del parametro μ .

Il metodo utilizzato per la stima della potenza del segnale usa una finestra rettangolare mobile:

$$\widehat{P}_x(n) = \frac{1}{M} \sum_{m=0}^{M-1} x^2(n-m) = \widehat{P}_x(n-1) + \frac{x^2(n) - x^2(n-M)}{M} \quad (1.11)$$

dove M è la durata della finestra rettangolare mobile. Questo metodo risulta una buona stima della potenza di un segnale.

Supponiamo di utilizzare una finestra rettangolare di durata pari al numero di coefficienti del filtro. Definiamo il vettore di ingresso all'istante k come:

$$\mathbf{x}(k) := [x(k), x(k-1), \dots, x(k-L+1)]^T$$

In questo caso risulta:

$$\widehat{P}_x(k) = \frac{\mathbf{x}^T(k) \mathbf{x}(k)}{M} \quad (1.12)$$

e quindi l'equazione 1.10 diviene:

$$\mu(k) = \frac{\alpha}{\mathbf{x}^T(k) \mathbf{x}(k)} \quad (1.13)$$

Di seguito vengono esposti alcuni accorgimenti per l'implementazione dell'algoritmo forniti in [1]:

1. scegliere $\widehat{P}_x(0)$ come migliore stima *a priori* della potenza del segnale di ingresso;
2. dato che non è auspicabile che la potenza stimata $\widehat{P}_x(k)$ sia nulla o molto piccola, si impone un vincolo al software implementato come segue:

$$\mu(k) = \frac{\alpha}{L \cdot \max \left\{ \widehat{P}_x(k), P_{min} \right\}}$$

dove P_{min} rappresenta il limite inferiore della potenza. Questa equazione assicura che $\mu(k)$ sia limitato anche se $\widehat{P}_x(k)$ dovesse risultare molto piccola quando il segnale risulta assente per un lungo periodo.

Il codice sorgente in linguaggio MATLAB del filtro che utilizza il presente algoritmo è fornito in Appendice A.2.

1.3.2. Algoritmo *Leaky LMS*

Un'insufficiente sollecitazione spettrale del segnale può dare luogo ad una divergenza del vettore dei pesi. Questo può accadere, per esempio, con una sinusoida non rumorosa quando si hanno più di due coefficienti. In questo caso si ottiene una crescita incontrollata del vettore dei pesi.

Ciò può essere spesso evitato mediante un “meccanismo di perdita” usato durante il calcolo per l'aggiornamento dei coefficienti.

In pratica l'algoritmo *Leaky LMS* viene rappresentato come segue:

$$\mathbf{w}(k+1) = \nu \mathbf{w}(k) + \mu \mathbf{x}(k)e(k) \quad (1.14)$$

dove ν è il fattore di perdita che rispetta alla condizione $0 < \nu \leq 1$. L'aggiunta di questo fattore di perdita porta ad una degradazione delle performance del filtro. Il valore del fattore di perdita viene in generale determinato dal costruttore su basi sperimentali tenendo presente che dev'essere un compromesso tra la robustezza e la perdita di performance.

1.4. Ambiti applicativi

Le principali caratteristiche ricercate in un filtro adattativo sono la capacità di operare all'interno di sistemi che hanno un comportamento sconosciuto e quella di inseguire le variazioni temporali di questi sistemi o segnali in ingresso. Queste caratteristiche fanno dei filtri adattativi uno strumento molto potente nei settori dell'elaborazione del segnale e del controllo.

La differenza sostanziale tra le varie applicazioni possibili è *da dove* viene estratto il segnale in ingresso. Generalmente ci sono quattro grandi rami d'applicazione dei filtri adattativi:

- cancellazione del rumore
- identificazione del sistema
- modellizzazione inversa
- predizione

Verranno brevemente illustrati solamente i primi due, in quanto strettamente legati tra loro e di interesse per questa tesi.

1.4.1. Cancellazione adattativa del rumore

Il principio di funzionamento della tecnica di cancellazione adattativa del rumore è mostrato in figura 1.3. La tecnica si basa sull'elaborazione del segnale in arrivo da due diversi sensori e ha come scopo la riduzione del rumore non desiderato attraverso un filtro adattativo.

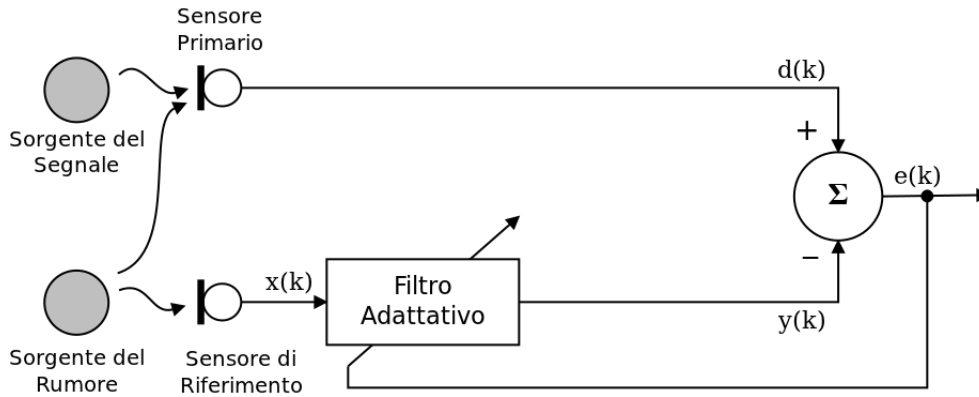


Figura 1.3.: Principio di funzionamento della cancellazione adattativa del rumore

Il sensore primario è collocato in prossimità del segnale sorgente al fine di catturare il segnale desiderato, tuttavia capta anche il segnale in arrivo dalla sorgente del rumore. Il sensore di riferimento, invece, è collocato in prossimità della sorgente di rumore al fine di catturare il segnale di rumore stesso.

Il diagramma a blocchi di un sistema di cancellazione adattativa del rumore è rappresentato in figura 1.4 dove $P(z)$ rappresenta la funzione di trasferimento tra la sorgente di rumore e il sensore primario e $W(z)$ rappresenta la funzione di trasferimento del filtro adattativo.

Il sistema ha due ingressi: l'ingresso primario $d(k)$ e l'ingresso di riferimento $x(k)$.

L'ingresso primario $d(k)$ consiste nella somma di due segnali distinti, $s(k)$ e $x'(k)$. L'obiettivo del filtro adattativo è di usare il riferimento in ingresso $x(k)$ per stimare il rumore $x'(k)$. L'uscita del filtro $y(k)$, che è una stima del rumore $x'(k)$, viene poi sottratta al segnale di ingresso primario $d(k)$. Nelle applicazioni ANC il segnale $s(k)$ non è presente e l'intento è quello di cancellare completamente il segnale primario.

Per minimizzare il rumore residuo $e(k) = d(k) - y(k)$ il filtro adattativo $W(z)$ genera in uscita un segnale $y(k)$ che altro non è che l'approssimazione di

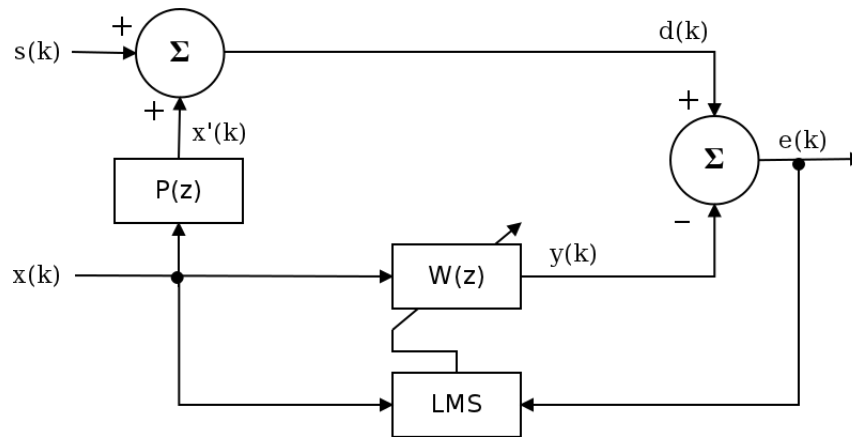


Figura 1.4.: Diagramma a blocchi di un sistema di cancellazione adattativa del rumore.

$x'(k)$. Quindi il filtro adattativo converge al sistema incognito $P(z)$ cioè esegue un'identificazione del sistema che sarà discussa nel seguente paragrafo.

1.4.2. Identificazione adattativa del sistema

Per identificazione si intende la ricostruzione del modello di un sistema a partire da una serie di dati raccolti. L'idea di fondo è di misurare i segnali prodotti dal sistema e di usarli per costruire un modello del sistema stesso.

L'identificazione di un sistema può avvenire in due modi a seconda di quando si esegue l'identificazione:

off-line identification: l'identificazione del modello avviene una volta acquisiti tutti i dati;

on-line identification: l'identificazione del modello avviene *on-line* cioè nel medesimo momento in cui si stanno raccogliendo i dati, ovvero il modello viene identificato e aggiornato istante per istante ogni qual volta si renda disponibile un nuovo dato. Questo tipo di identificazione viene anche detta identificazione adattativa.

Un esempio di identificazione adattativa di un sistema è illustrato in figura 1.5, dove $P(z)$ rappresenta il sistema incognito⁵ che deve essere identificato e $W(z)$ rappresenta il filtro digitale usato per modellizzare $P(z)$.

⁵Per sistema incognito si intende un sistema, un processo o un impianto di cui non si conoscono le caratteristiche "interne" ma solamente i segnali che fornisce in uscita attraverso l'applicazione di un segnale in ingresso.

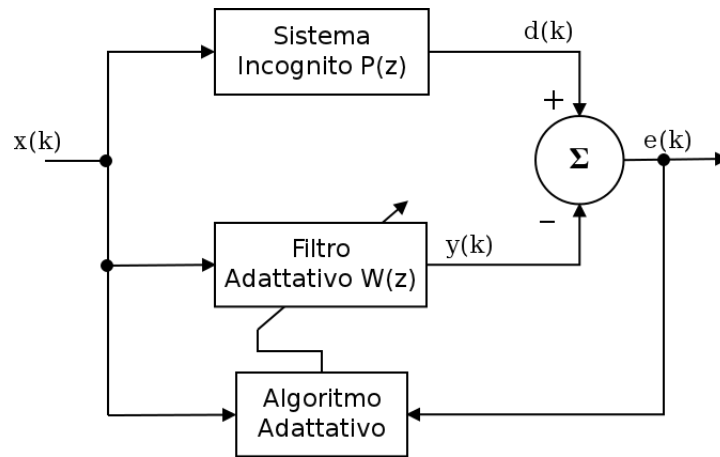


Figura 1.5.: Identificazione adattativa di un sistema

$W(z)$ si appoggia ad un algoritmo adattativo per la minimizzazione di una funzione di errore predeterminata. Infatti se il filtro adattativo $W(z)$ viene sollecitato in ingresso dal medesimo segnale di $P(z)$, cioè $x(k)$, misurando i rispettivi segnali di uscita $y(k)$ e $d(k)$ si può risalire alle caratteristiche di $P(z)$ facendo variare i coefficienti del filtro digitale, ovvero minimizzando la differenza tra i due segnali in uscita $e(k) = y(k) - d(k)$. Questa tecnica viene ampiamente utilizzata in due grandi aree: quella della cancellazione del rumore e quella della cancellazione degli echi.

Capitolo 2.

Controllo attivo feedforward del rumore a banda larga

Per controllo *feedforward* si intende un tipo di controllo nel quale parte dei disturbi che incidono sul sistema possono essere misurati e utilizzati dal controllore dando luogo a quella che viene comunemente detta azione di compensazione a catena aperta.

Nel presente capitolo prenderemo in considerazione una classe di sistemi formata da:

- un singolo sensore di riferimento
- una singola sorgente secondaria
- un singolo sensore di errore

Un esempio di questa classe di sistema è illustrato in figura 2.1 e può essere descritto come un problema di identificazione di sistema.

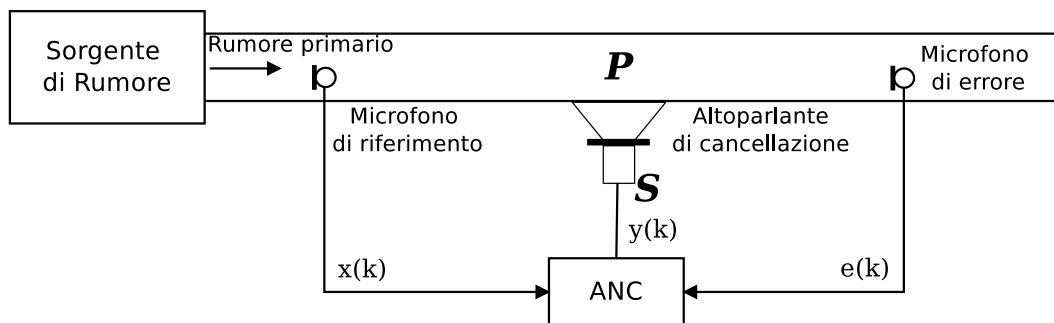


Figura 2.1.: Sistema ANC a banda larga in controreazione a singolo canale.

Il rumore non desiderato proveniente dalla sorgente primaria viene misurato attraverso un microfono, detto microfono di riferimento, viene poi filtrato attraverso un filtro adattativo e usato per pilotare una seconda sorgente sonora, ovvero un altoparlante detto altoparlante di cancellazione, al fine di tentare di cancellare il rumore all'interno del condotto.

Nei sistemi che controllano un rumore (casuale) a banda larga, il segnale di riferimento fornisce al filtro adattativo delle informazioni riguardanti le caratteristiche del rumore primario *prima* che esso raggiunga l'altoparlante di cancellazione. Questo è un requisito necessario ai fini di un controllo causale. Il rimanente rumore viene poi catturato da un secondo microfono, detto microfono di errore, e usato per l'aggiornamento dei coefficienti del filtro adattativo allo scopo di minimizzare il rumore residuo stesso.

L'ipotesi che ora viene fatta è che il segnale di riferimento non sia affetto da un'interferenza dovuta al "ritorno" del segnale generato dall'altoparlante.

Uno schema a blocchi semplificato di un sistema di questo tipo è mostrato in figura 2.2¹.

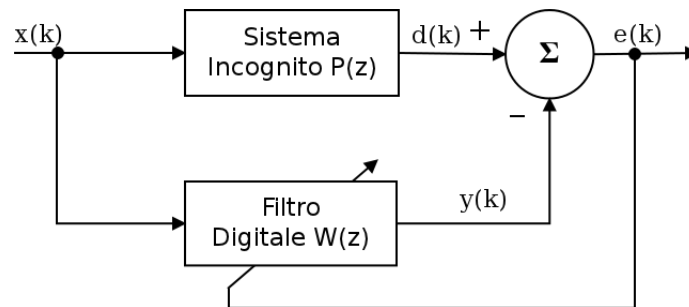


Figura 2.2.: ANC: problema di identificazione del sistema.

Questo tipo di problema rientra nella classe "identificazione del sistema": in questo caso un filtro adattativo $W(z)$ è usato per stimare il sistema incognito $P(z)$. Sia il sistema incognito che il filtro adattativo sono pilotati dal medesimo segnale in ingresso $x(k)$. L'uscita del sistema incognito risulta essere la risposta "desiderata" del filtro adattativo. Se il sistema è tempo variante, il modello sarà tempo variante e l'algoritmo adattativo $W(z)$ avrà il compito di mantenere l'er-

¹Si vuole far notare che il blocco sommatore viene nella pratica realizzato attraverso una sovrapposizione (quindi da una somma) di due segnali acustici. Per semplicità si continuerà a rappresentare questa somma come una sottrazione di due segnali elettrici. Questo fatto è del tutto arbitrario in quanto esso può essere arbitrariamente realizzato mediante un semplice cambio di segno.

rore di modellizzazione il più piccolo possibile attraverso continue variazioni dei coefficienti del filtro, cioè dovrà minimizzare il segnale di errore residuo $e(k)$.

Dalla figura 2.2, calcoliamo la trasformata z del segnale $e(k)$:

$$E(z) = D(z) - Y(z) = P(z)X(z) - W(z)X(z)$$

Idealmente $E(z) = 0$ dopo la convergenza del filtro adattativo $W(z)$. Dall'equazione precedente si ottiene che:

$$W(z) = P(z)$$

prestando particolare attenzione al fatto che non devono esserci valori di z tali per cui $X(z)$ abbia valore nullo.

L'equazione precedente implica che:

$$y(k) = d(k)$$

condizione che rappresenta il fatto che l'uscita del filtro $y(k)$ è identica al disturbo primario $d(k)$. Quindi quando $d(k)$ e $y(k)$ sono acusticamente in perfetta sovrapposizione, il rumore residuo è:

$$e(k) = d(k) - y(k) = 0$$

portando ad una perfetta cancellazione del rumore.

Si capisce quindi che il modello rappresentato da $W(z)$ è una buona rappresentazione del sistema incognito $P(z)$ e quindi che l'approccio feedforward risulta in grado di cancellare un rumore casuale a banda larga in maniera efficace. Chiaramente bisogna assumere come ipotesi che l'algoritmo adattativo abbia tempo sufficiente per:

1. la computazione
2. la determinazione del segnale inverso
3. la generazione dello stesso *prima* che il segnale primario passi attraverso il sistema.

2.1. Effetto del percorso secondario

L'applicazione pratica dei filtri adattativi comporta alcune complicazioni. Infatti l'utilizzo dell'algoritmo LMS introdotto nel capitolo 1, presuppone che il segnale di errore sia disponibile come differenza tra il disturbo primario e l'uscita del filtro adattativo, ma ciò non accade nella pratica. Inoltre l'utilizzo di un filtro adattativo come quello mostrato in figura 2.2 all'interno di un sistema come quello in figura 2.1 è complicato dal fatto che il segnale (elettrico) di riferimento deve essere ottenuto a partire dalla pressione acustica esercitata dal rumore attraverso un microfono. Allo stesso modo il segnale di errore (elettrico) deve essere ottenuto per mezzo di un microfono a partire dal rumore (acustico) residuo. Infine il rumore di cancellazione (*anti-noise*) deve essere prodotto attraverso un altoparlante. Si può intuire quindi come all'interno dello schema a blocchi vadano aggiunte una serie di ulteriori funzioni di trasferimento al fine di modellizzare i comportamenti dinamici dei microfoni e dell'altoparlante ottenendo quindi una rappresentazione come quella di figura 2.3 dove il nodo sommatore rappresenta la sovrapposizione, nel dominio acustico, del segnale prodotto dall'altoparlante di cancellazione con il rumore primario nello spazio compreso tra l'altoparlante e il microfono di errore.

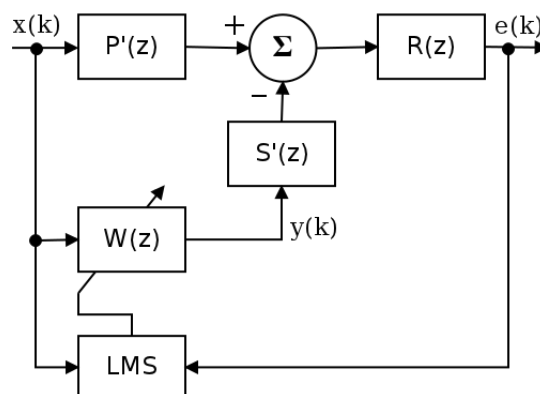


Figura 2.3.: Diagramma a blocchi di un sistema ANC

Risulta quindi necessario compensare la funzione di trasferimento del percorso secondario $S(z)$ aggiungendo una serie di blocchi all'interno dello schema. Come illustrato in figura 2.3, la funzione di trasferimento del percorso secondario $S(z)$ può essere separata in due blocchi posti in cascata tra di loro:

$$S(z) = R(z)S'(z)$$

dove $S'(z)$ rappresenta la funzione di trasferimento secondaria dall'uscita del filtro adattativo al blocco sommatore e $R(z)$ rappresenta la funzione di trasferimento residua dal blocco sommatore al segnale di errore. Una rappresentazione di $S'(z)$ può essere il sistema di altoparlanti (comprensivo di convertitore D/A, amplificatore, ecc...) mentre $R(z)$ rappresenta la funzione di trasferimento del sistema microfonico, comprensivo anch'esso di eventuali amplificatori, convertitori A/D, ecc...

Allo stesso modo anche la funzione di trasferimento del segnale di ingresso $P(z)$ può essere separata come:

$$P(z) = R(z)P'(z)$$

dove $P'(z)$ è la funzione di trasferimento del sistema acustico incognito dal microfono di riferimento al blocco sommatore.

I coefficienti del filtro adattativo devono convergere ad un valore adeguato che minimizzi il segnale di errore nonostante la presenza di tutte queste funzioni di trasferimento aggiuntive. Dallo schema di figura 2.3 si evince che la trasformata Z del segnale di errore risulta essere:

$$E(z) = R(z)[P'(z) - S'(z)W(z)]X(z)$$

Assumendo che $W(z)$ abbia ordine sufficiente si ottiene, dopo la convergenza del filtro adattativo, che l'errore residuo sia nullo (cioè $E(z) = 0$). Questo richiede che $W(z)$ realizzi la funzione di trasferimento ottimale:

$$W_{opt}(z) = \frac{P'(z)}{S'(z)} \quad (2.1)$$

Per fare ciò il filtro adattativo deve invertire la funzione di trasferimento del percorso secondario $S'(z)$. Da notare che questa operazione può portare all'instabilità del sistema ad una certa pulsazione ω tale che $S'(e^{j\omega}) = 0$.

Dall'equazione 2.1 si può vedere che il filtro adattativo $W(z)$ esegue una modellizzazione diretta della funzione di trasferimento $P'(z)$ per generare il segnale di cancellazione e una modellizzazione inversa di $S'(z)$ per compensare l'effetto del percorso secondario. Dato che sia la funzione di trasferimento primaria che la funzione di trasferimento secondaria hanno $R(z)$ in comune è conveniente aggregare questi termini nel modello. Ciò porta ad una rappresentazione come quella

di figura 2.4 la quale risulta esattamente equivalente allo schema di figura 2.3 con la sola differenza che il blocco sommatore viene ridefinito nel dominio elettrico anziché nel dominio acustico.

Le funzioni di trasferimento del filtro adattativo possono quindi essere ridefinite risultando di più semplice comprensione. La trasformata z del segnale diviene:

$$E(z) = [P(z) - S(z)W(z)]X(z)$$

Anche in questo caso, dopo la convergenza del filtro adattativo, l'errore residuo è nullo, cioè $E(z) = 0$, il che implica la necessità di $W(z)$ di realizzare la funzione di trasferimento ottimale:

$$W_{opt}(z) = \frac{P(z)}{S(z)} \quad (2.2)$$

In altre parole, il filtro adattativo $W(z)$ modella direttamente $P(z)$ e inversamente $S(z)$.

Il filtro adattativo $W(z)$ offre una modalità completamente adattabile simultaneamente alla funzione di trasferimento primaria del sistema, $P(z)$, e alla funzione di trasferimento secondaria, $S(z)$, che minimizza il rumore residuo. Il vantaggio di questo approccio sta nel fatto che con un modello appropriato di $P(z)$, il sistema può rispondere istantaneamente ai cambiamenti del segnale di ingresso causati da variazioni della sorgente del rumore.

Nei casi di controllo attivo del rumore $S(z)$ viene introdotta dopo il filtro adattativo, come mostrato in figura 2.4. In questi casi risulta necessaria una modifica dell'algoritmo LMS introdotto nel capitolo 1 al fine di assicurarne la convergenza, il che porta ad ottenere l'algoritmo FXLMS (*Filtered-X LMS Algorithm*) presentato nella seguente sezione.

2.2. Algoritmo FXLMS

Come già accennato in precedenza, nel caso in cui la funzione di trasferimento del percorso secondario $S(z)$ sia a valle del filtro adattativo $W(z)$, l'algoritmo LMS nella sua forma tradizionale non può essere impiegato ma necessita di una modifica in quanto bisogna compensare l'effetto dell'aggiunta della funzione di trasferimento $S(z)$.

Esistono molteplici soluzioni per compensare tale effetto:

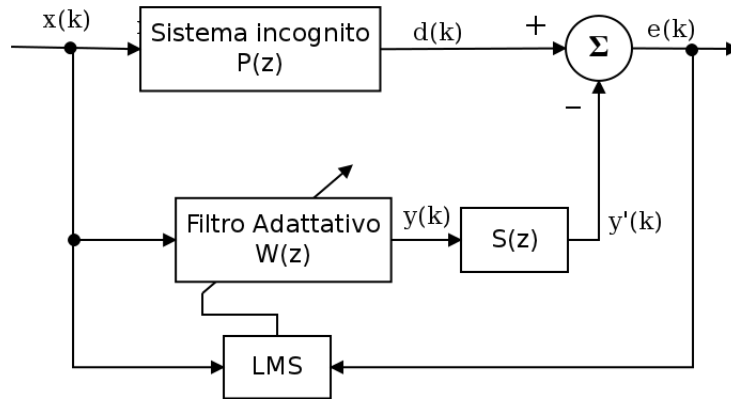


Figura 2.4.: Diagramma a blocchi semplificato di un sistema ANC

- la prima soluzione è quella di piazzare un filtro inverso, $1/S(z)$, in serie con $S(z)$ per rimuoverne l'effetto.
- la seconda, che verrà esposta nella presente sezione, è quella di inserire un filtro identico a $S(z)$ nel percorso del segnale di riferimento che aggiorna i coefficienti del filtro adattativo pilotato dall'algoritmo LMS. Questa soluzione viene realizzata, per l'appunto, dall'algoritmo FXLMS o *Filtered-X LMS algorithm*.

Quest'ultimo approccio risulta migliore rispetto al primo in quanto non necessita del calcolo dell'inversa della funzione $S(z)$. Bisogna però imporre una condizione di tempo invarianza per il filtro adattativo $W(z)$ ovvero bisogna supporre che il comportamento del filtro sia tempo invariante o variabile lentamente, condizione assunta per vera in questa tesi.

2.2.1. Derivazione dell'algoritmo

Basandosi sullo schema mostrato in figura 2.4, risulta evidente che il segnale di errore residuo può essere espresso come:

$$\begin{aligned}
 e(k) &= d(k) - y'(k) \\
 &= d(k) - [s * y](k) \\
 &= d(k) - [s * (\mathbf{w}^T \cdot \mathbf{x})](k)
 \end{aligned} \tag{2.3}$$

dove $s(k)$ indica la risposta impulsiva del percorso secondario $S(z)$ al tempo k , il simbolo $*$ indica l'operazione di convoluzione,

$$\mathbf{w}(k) = [w_0(k), w_1(k) \dots w_{L-1}(k)]^T$$

è il vettore corrispondente a $W(z)$ con i relativi coefficienti al tempo k ,

$$\mathbf{x}(k) = [x(k), x(k-1), \dots x(k-L+1)]^T$$

è il vettore del segnale al tempo k ed L è l'ordine del filtro $W(z)$.

Per segnali a banda larga $w(k)$ rappresenta la risposta impulsiva della funzione di trasferimento fornita dall'equazione (2.2).

L'obiettivo del filtro adattativo è quello di minimizzare l'errore quadratico istantaneo, $\hat{J}(k) = e^2(k)$ e come già discusso nel capitolo 1, il metodo più usato per ottenerlo è l'algoritmo LMS, che aggiorna i coefficienti del vettore del filtro nella direzione del gradiente negativo con fattore di convergenza μ . Tale metodo è esplicitato nella formula seguente:

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \frac{\mu}{2} \nabla \hat{J}(k) \quad (2.4)$$

$\nabla \hat{J}(k)$ rappresenta la stima istantanea del gradiente dell'errore quadratico istantaneo al tempo k ed è espresso come:

$$\nabla \hat{J}(k) = \nabla e^2(k) = 2[\nabla e(k)]e(k)$$

Dall'equazione (2.3) si ottiene:

$$\nabla e(k) = -[s * \mathbf{x}](k) = -\mathbf{x}'(k)$$

dove

$$\mathbf{x}'(k) = [x'(k) x'(k-1) \dots x'(k-L+1)]^T$$

e

$$x'(k) = [s * x](k)$$

Pertanto la stima del gradiente risulta essere:

$$\nabla \hat{J}(k) = -2\mathbf{x}'(k)e(k)$$

che sostituita nell'equazione (2.4) porta alla forma finale dell'algoritmo FXLMS:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu \mathbf{x}'(k)e(k) \quad (2.5)$$

Questo risultato mostra che quando una funzione di trasferimento, nel nostro caso $S(z)$, è posta a valle del filtro adattativo deve essere piazzata anche nel percorso di aggiornamento dei coefficienti del filtro stesso: da qui il termine “*filtered-X*”.

Nella pratica la conoscenza di $S(z)$ non è immediata. Solitamente per assicurare la convergenza dell'algoritmo, nel percorso secondario viene posto un filtro che risulta essere una *stima* della funzione di trasferimento $S(z)$ originaria. A causa dell'introduzione di questa funzione stimata, l'algoritmo va opportunamente corretto. Esso risulta comunque rappresentato dall'equazione (2.5) anche se bisogna correggere la definizione di $\mathbf{x}'(k)$ modificandola in:

$$\mathbf{x}'(k) = [\hat{s} * x](k)$$

Lo schema a blocchi dell'algoritmo, comprensivo di funzione stimata $\hat{S}(z)$, è rappresentato in figura 2.5.

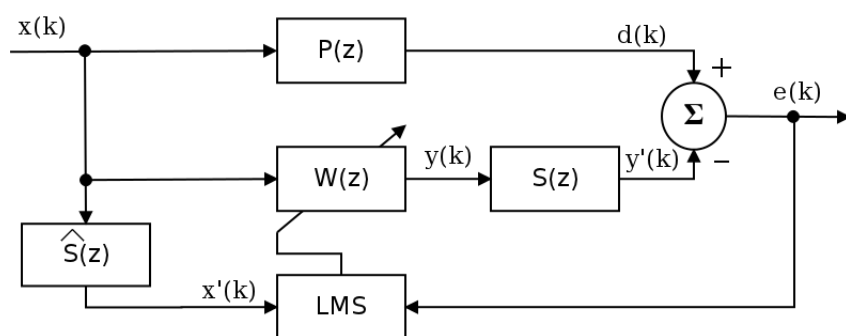


Figura 2.5.: Diagramma a blocchi di un sistema ANC che utilizza l'algoritmo FXLMS

In questa tesi le simulazioni riguardanti l'algoritmo FXLMS sono state fatte in due modi distinti. La discriminante è stata *come* si è scelto di implementare la funzione di trasferimento stimata $\hat{S}(z)$ ovvero in base a quanto bene si è ipotizzato di conoscere la funzione di trasferimento $S(z)$. Dato che nella pratica risulta impossibile conoscere con esattezza tale funzione di trasferimento, si è scelto di mostrare come le prestazioni dell'algoritmo variano rispetto ad un caso ideale dove si ipotizza di conoscere perfettamente la funzione di trasferimento del percorso secondario.

In altre parole i due casi in esame sono:

1. conoscenza esatta di $S(z)$. In questo caso (ideale) si è ipotizzato di avere una perfetta conoscenza della funzione di trasferimento $S(z)$ e quindi di poterla utilizzare direttamente in sostituzione a $\widehat{S}(z)$ stessa.
2. stima adattativa di $S(z)$. In questo caso si è usato il toolbox `ident`, presente in MATLAB, per stimare la funzione di trasferimento $S(z)$.

Le simulazioni dell'algoritmo FXLMS con le varie scelte effettuate saranno oggetto dei prossimi paragrafi.

2.2.1.1. Tecnica di modellizzazione off-line

Si assuma che le caratteristiche della funzione di trasferimento $S(z)$ siano tempo-invarianti ma sconosciute. La tecnica di modellizzazione off-line può essere usata per stimare $S(z)$ durante una fase iniziale di analisi. Alla fine di questa fase la stima $\widehat{S}(z)$ ottenuta viene fissata e usata nel funzionamento dell'algoritmo. Un buon segnale di test, che viene utilizzato in questo caso, è un rumore bianco, ovvero un segnale caratterizzato da una densità spettrale di potenza costante su tutto l'orizzonte frequenziale. Il rumore bianco risulta essere un buon segnale di test perché eccita il sistema in un ampio spettro di frequenze. Per generare questo rumore bianco si usa un generatore di rumore casuale che viene poi posto in ingresso ad un filtro adattativo $\widehat{S}(z)$ usato per modellizzare il percorso secondario.

La procedura che riassume la tecnica di modellizzazione off-line è presentata di seguito. Per la nomenclatura dei segnali ci si riferisce alla figura 2.5:

1. Generare un rumore bianco $x(k)$.
2. Ottenere il segnale desiderato $d(k)$ dal sensore di errore.
3. Applicare l'algoritmo di filtraggio adattativo come segue:
 - a) calcolare l'uscita del filtro adattativo

$$y(k) = \sum_{l=0}^{L-1} \widehat{s}_l(k)x(k-l)$$

dove $\widehat{s}_l(k)$ rappresenta l -esimo coefficiente del filtro del percorso secondario stimato $\widehat{S}(z)$ al tempo k ;

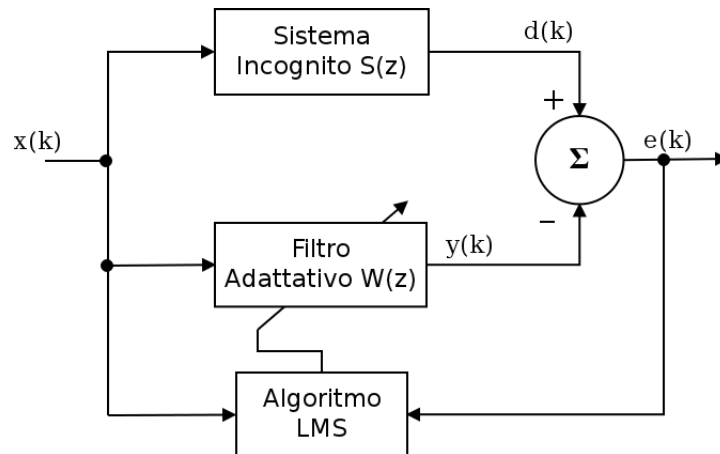


Figura 2.6.: Modellizzazione di un sistema incognito tramite LMS

b) calcolare il segnale di errore

$$e(k) = d(k) - y(k)$$

c) aggiornare i coefficienti utilizzando il classico algoritmo LMS

$$\hat{s}_l(k+1) = \hat{s}_l(k) + \mu x(k-l)e(k), \quad l = 0, 1, \dots, L-1$$

4. Tornare al punto 1 per l'iterazione successiva finché il filtro adattativo $\hat{S}(k)$ converge alla soluzione ottimale, cioè finché la potenza del segnale di errore $e(k)$ non viene minimizzata.

Dopo la convergenza dell'algoritmo i coefficienti \hat{s}_l , $l = 0, 1, \dots, L-1$ verranno utilizzati nel sistema di controllo attivo².

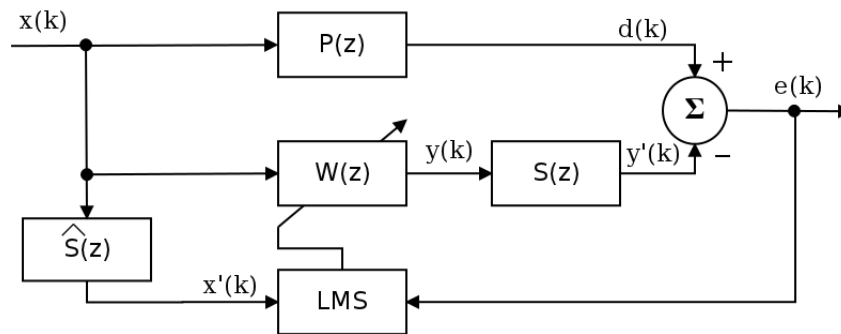
Da notare che questa tecnica, sebbene risulti utilizzabile nelle applicazioni di controllo attivo del rumore, ha il grosso svantaggio di non riuscire ad adattarsi ai cambiamenti $S(z)$ nel corso del funzionamento ma richiede un'ulteriore fase di analisi per poter stimare nuovamente $S(z)$.

²Si vuole far notare che una operazione di media su ripetuti calcoli comporterebbe un miglioramento di tale risultato

2.3. Simulazioni in Matlab di FXLMS

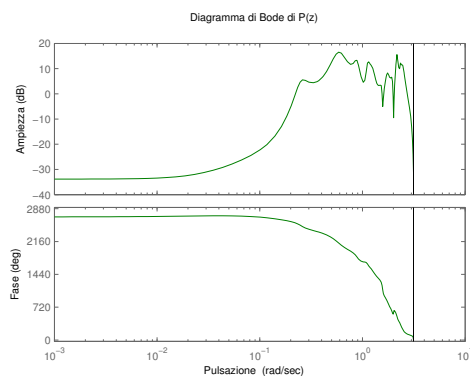
Nelle seguenti simulazioni si vuole mostrare come si comporta l'algoritmo FXLMS introducendo un'operazione di stima della funzione di trasferimento $S(z)$. Si vuole far notare che l'utilizzo di una versione stimata della funzione di trasferimento $S(z)$ comporta l'insorgere di un certo errore dovuto all'operazione di stima stessa.

Per la nomenclatura si farà riferimento alla figura 2.5 riportata di seguito per comodità.

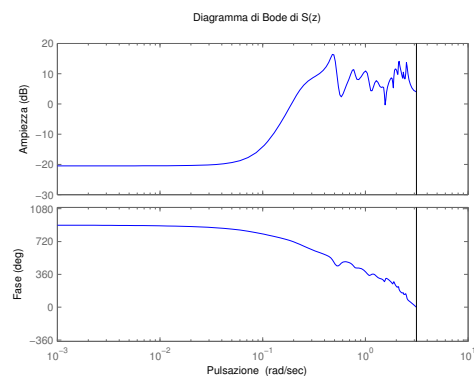


I dati in ingresso scelti per la simulazione sono:

1. funzione di trasferimento primaria $P(z)$ del ventiquattresimo ordine. Il diagramma di Bode di $P(z)$ è riportato in figura 2.7a;
2. funzione di trasferimento secondaria $S(z)$ del ventiquattresimo ordine. Il diagramma di Bode di $S(z)$ è riportato in figura 2.7b;



(a) Diagramma di Bode di $P(z)$



(b) Diagramma di Bode di $S(z)$

Figura 2.7.: Rappresentazioni delle funzioni di trasferimento primaria e secondaria

3. il rumore di ingresso $x(k)$ è stato prodotto partendo da un rumore bianco che è stato fatto passare per un filtro avente una certa pulsazione di risonanza $\omega_r = 250 [mHz]$; Lo spettro del segnale in uscita è visibile in figura 2.8.
4. il filtro adattativo $W(z)$ è stato scelto con i seguenti parametri:
 - a) $L = 64$
 - b) $\mu = 0.001$

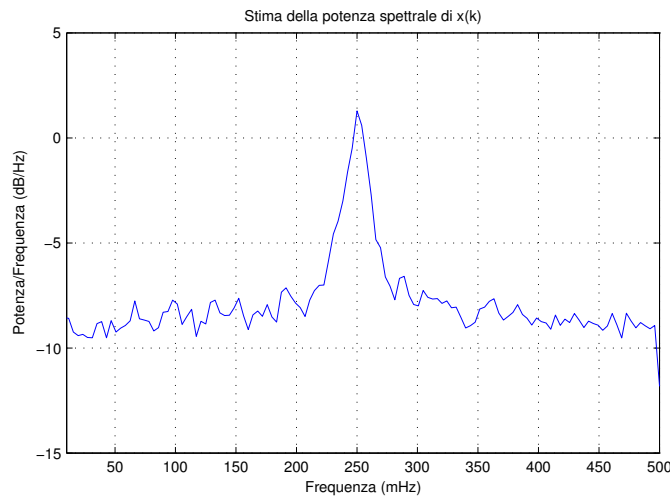


Figura 2.8.: Spettro del segnale $x(k)$.

2.3.1. Primo caso: conoscenza esatta di $S(z)$

In questo caso si è utilizzata come $\hat{S}(z)$ la funzione di trasferimento $S(z)$, simulando così una perfetta conoscenza della funzione di trasferimento del percorso secondario (caso ideale).

Onde evitare di ottenere un grafico poco interpretabile, si è preferito graficare solamente l'andamento di *alcuni* elementi del vettore dei pesi per mostrarne la convergenza. L'andamento di questi è riportato in figura 2.9.

Il segnale di errore $e(k)$ è mostrato, assieme al segnale di riferimento $d(k)$, in figura 2.10. Il segnale residuo risulta attenuato rispetto al segnale di ingresso all'incirca di $18 dB$.

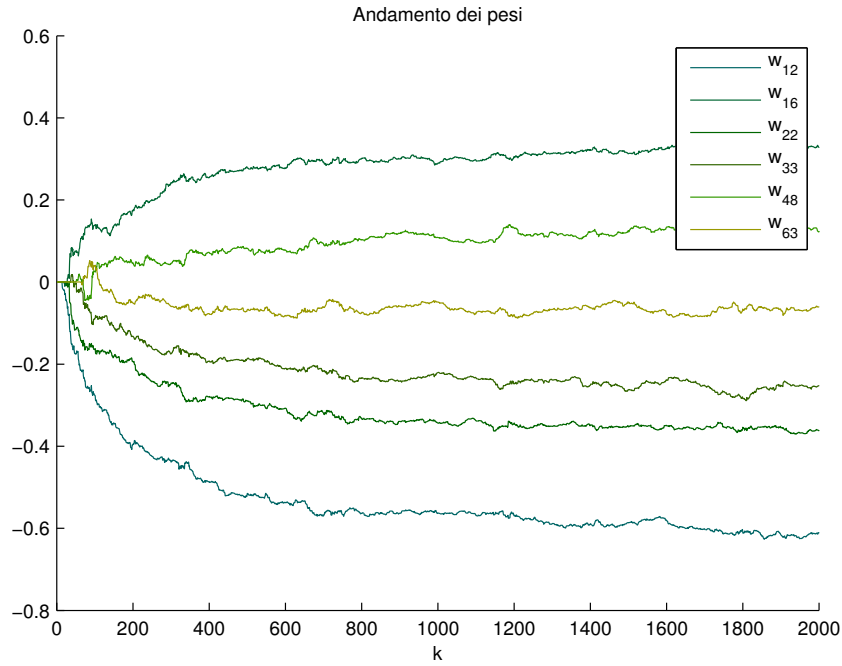


Figura 2.9.: Andamento dei pesi w_{12} , w_{16} , w_{22} , w_{33} , w_{48} , w_{63} in un intervallo da $k = 0$ a $k = 2000$ istanti

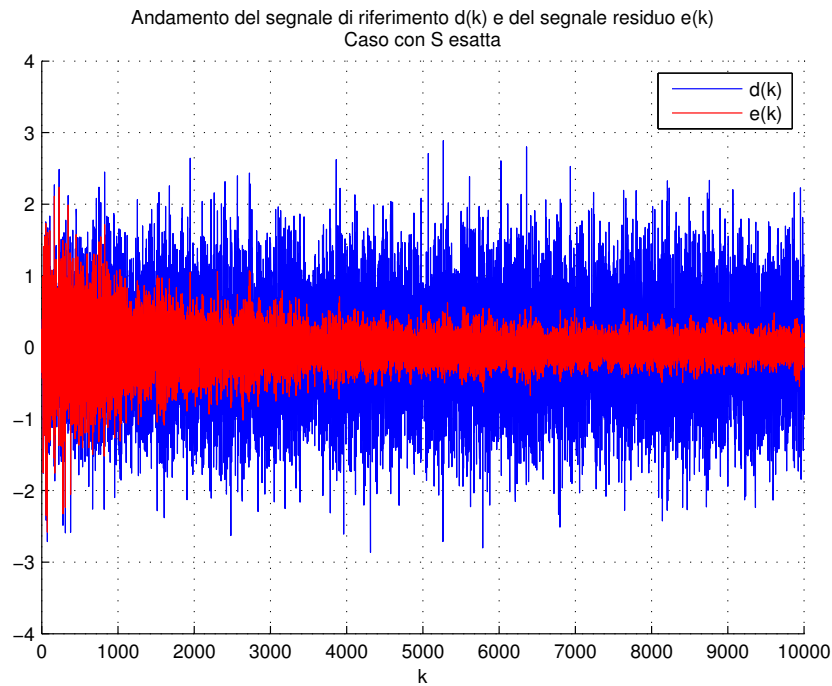


Figura 2.10.: Confronto tra il segnale di riferimento $d(k)$ e il segnale di errore residuo $e(k)$.

2.3.2. Secondo caso: stima di $S(z)$

In questo secondo caso si è provveduto ad ottenere una stima della funzione di trasferimento secondaria $S(z)$ attraverso il toolbox `ident` presente in MATLAB. Per fare ciò si è generato un rumore bianco e si è analizzata, tramite il suddetto toolbox, la risposta di $S(z)$ a tale segnale. La funzione di trasferimento stimata $\hat{S}(z)$ è stata poi usata per filtrare il segnale in ingresso $x(k)$ prima che esso sia posto in ingresso all'algoritmo adattativo LMS.

Come modello di stima si è scelto un modello parametrico lineare di tipo auto-regressivo (ARX). I parametri sono stati impostati al fine di ottenere una funzione di trasferimento al massimo del quindicesimo ordine. La scelta è stata fatta al fine di ottenere un certo errore nella stima della funzione di trasferimento dovuto alla mancata conoscenza esatta di $S(z)$.

In figura 2.11 sono mostrati i diagrammi di Bode della funzione di trasferimento $S(z)$ e della sua versione stimata $\hat{S}(z)$ ottenuta con le impostazioni discusse in precedenza.

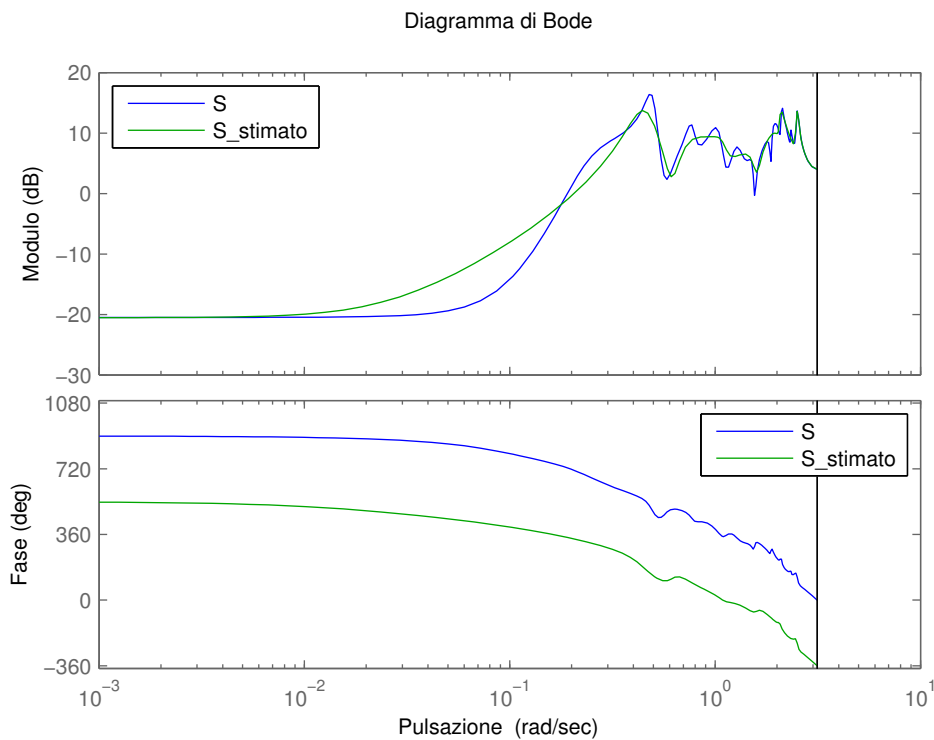


Figura 2.11.: Andamento delle funzioni di trasferimento $S(z)$ e $\hat{S}(z)$.

Notiamo dal grafico di figura 2.12 che l'andamento dei segnali è molto simile a quello di figura 2.10, infatti anche in questo caso si ha un'attenuazione tra il segnale di ingresso e il segnale residuo di circa 18 dB . Questo accade perché l'algoritmo LMS si adatta al fine di minimizzare l'errore complessivo che tiene conto anche della non conoscenza esatta di $S(z)$. In altre parole l'algoritmo adattativo si prende carico di compensare anche l'errore dovuto alla conoscenza imprecisa della funzione di trasferimento del percorso secondario.

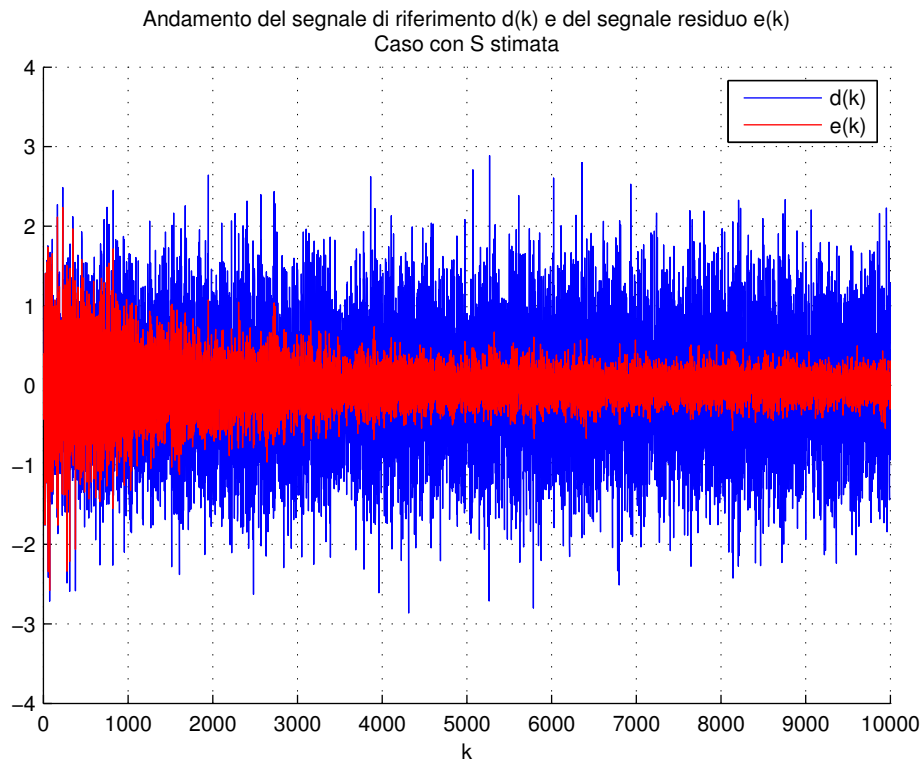


Figura 2.12.: Confronto tra il segnale di riferimento $d(k)$ e il segnale di errore residuo $e(k)$.

Capitolo 3.

ANC - Applicazione ai caschi per motoveicoli

Gli ultimi decenni hanno visto un aumento sproporzionato del traffico causato dalla sempre più consistente presenza su strada di automobili e autoveicoli per uso commerciale. Oltre a ciò, la tendenza a diminuire drasticamente i posti per il parcheggio, soprattutto nei centri città, ha contribuito a far sì che i motocicli non vengano più visti solamente come accessori per il divertimento del fine settimana ma come mezzi di trasporto agili e adatti ad affrontare questi inconvenienti con il pregio inoltre di ridotti consumi che portano ad avere anche un non indifferente vantaggio economico.

Tuttavia bisogna prendere in considerazione anche uno dei principali difetti derivanti dall'utilizzo di questi mezzi: il rumore percepito attraverso il casco. Questo non solo distoglie la concentrazione del pilota e porta ad una degradazione della sensazione di comfort, ma può anche portare, nel caso di un uso professionale, come per esempio per la polizia in motocicletta, ad un danno permanente dell'udito.

Anche se lo scopo principale dei caschi è quello di proteggere il pilota da eventuali cadute, esso deve assolvere anche ad altri compiti, come la protezione dalla pioggia, dal vento e dal rumore.

3.1. Natura del rumore percepito

Il rumore percepito normalmente da un motociclista durante la guida è costituito principalmente da:

- rumore generato dal motore (*engine noise*)

- rumore generato dal flusso d'aria (*wind noise*)

Oltre a questi, seppur in maniera minore, vi sono altri fattori che contribuiscono alla generazione del rumore quali ad esempio:

- pneumatici a contatto con l'asfalto
- condizioni del manto stradale

Il rumore percepito all'interno del casco dal pilota aumenta rapidamente all'aumentare della velocità. A velocità che si aggirano attorno ai 60-70 Km/h il rumore aerodinamico causato dalle turbolenze del flusso d'aria attorno al casco supera il rumore generato dal motociclo stesso.

Analisi sperimentali mostrano che il livello di rumore percepito da un motociclista su strada si aggira intorno a 80-90 dBA ad una velocità di circa 50 Km/h per arrivare fino a 116 dBA ad una velocità di circa 190 Km/h [3].

L'esposizione ad alti livelli di rumore per lungo tempo può distrarre la concentrazione del guidatore e rendere difficile la percezione di situazioni di pericolo, arrivando addirittura a provocare incidenti.

Il casco provvede all'attenuazione del rumore attraverso una fodera interna. La struttura risulta efficace nella riduzione del rumore a frequenze superiori ai 500Hz. Possiamo classificare questo tipo di attenuazione come un controllo passivo del rumore in quanto prevede l'utilizzo di barriere fisiche per limitare la propagazione del segnale acustico.

Riguardo l'attenuazione delle frequenze inferiori l'approccio migliore risulta quello del controllo attivo del rumore che riesce a cancellare efficacemente il segnale a basse frequenze.

Nella seguente sezione sarà considerata l'applicazione del controllo attivo feed-forward del rumore a banda larga discusso nel capitolo 2 ai caschi per motoveicoli.

3.2. Analisi del rumore

Il grafico spettrale del rumore è riportato in figura 3.1 e si riferisce ad una velocità di circa 60 Km/h.

Al fine di eseguire una simulazione attraverso MATLAB si è provveduto ad eseguire una registrazione audio di un rumore realistico. Per eseguire tale registrazione sono stati impiegati:

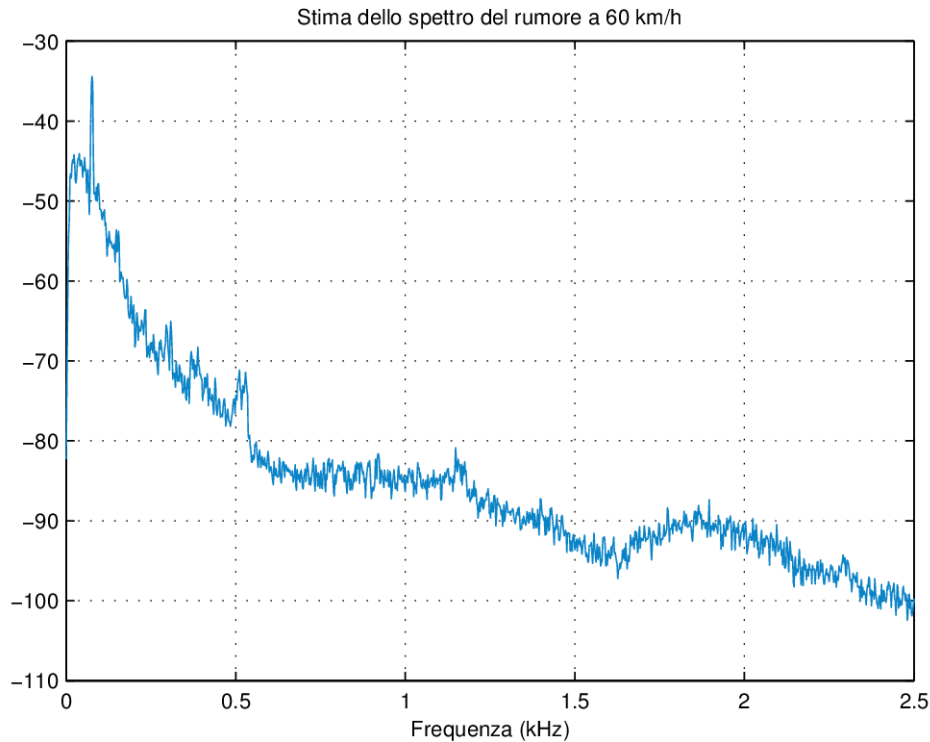


Figura 3.1.: Spettro del rumore ad una velocità di 60 Km/h

- una moto Yamaha FJ1200 del 1986 a quattro tempi con quattro cilindri;
- un casco Nolan N103;
- un registratore digitale mini-disc Sharp MD-MT877H(S);
- due microfoni e un altoparlante per interfono.

Il rumore prodotto dalla motocicletta è formato dal rumore periodico del motore e dal rumore d'ambiente dovuto per esempio a vento, pneumatici, strada, ecc... Esso può essere classificato in:

Rumore a bassa frequenza: è prodotto principalmente dal motore e può risultare molto fastidioso soprattutto nei lunghi tragitti.

Rumore a media frequenza: dovuto principalmente alla componentistica meccanica, come per esempio la trasmissione.

Rumore ad alta frequenza: dovuto generalmente alle turbolenze aerodinamiche.

I rumori a media ed alta frequenza vengono agevolmente ridotti da silenziatori di tipo resistivo (membrane) già presenti all'interno del casco, mentre per il rumore a bassa frequenza questa tecnica non consente di ottenere risultati accettabili.

Al fine di concentrarsi solo sul rumore a bassa frequenza, la registrazione è stata fatta ad una frequenza di campionamento $f_c = 8000 [Hz]$.

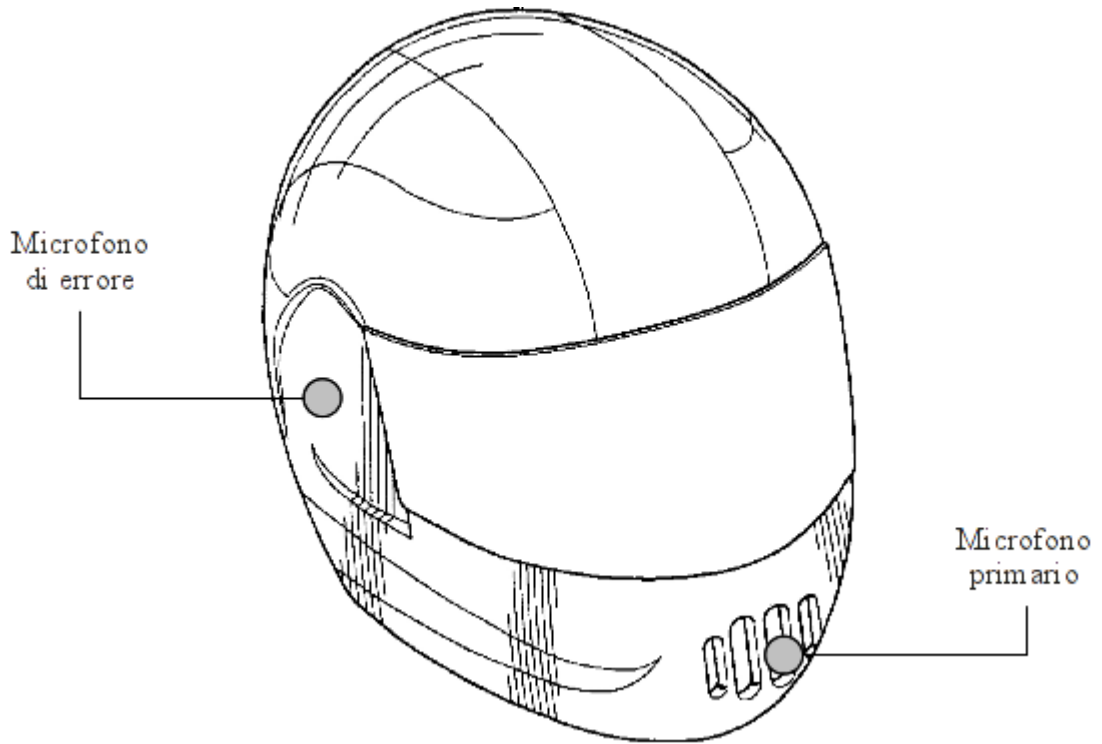
Come si può notare dalla figura, l'energia del rumore è concentrata al di sotto dei 500 Hz ma si sviluppa su un'ampia gamma di frequenze. Questo porta a classificare il rumore come *rumore a banda larga*.

3.2.1. Posizionamento dei sensori

I due microfoni usati per la registrazione sono stati posizionati all'interno del casco. Il microfono primario è stato posizionato vicino al mento del pilota, mentre il microfono secondario è stato piazzato vicino all'orecchio destro¹ come mostrato in figura 3.2a.

L'obiettivo del sistema ANC è quello di minimizzare il rumore residuo $e(k)$ rilevato dal microfono di errore creando una zona di minimo rumore vicino all'orecchio del pilota. Il posizionamento del microfono di errore risulta quindi un punto cruciale per la costruzione di un buon sistema ANC. In questo caso si è posizionato il microfono come in figura 3.2c poiché tale posizione risulta essere migliore rispetto alle altre per il funzionamento del sistema di cancellazione [4].

¹Si è scelto di utilizzare in questo set-up sperimentale un solo altoparlante



(a) Posizionamento dei microfoni all'interno del casco

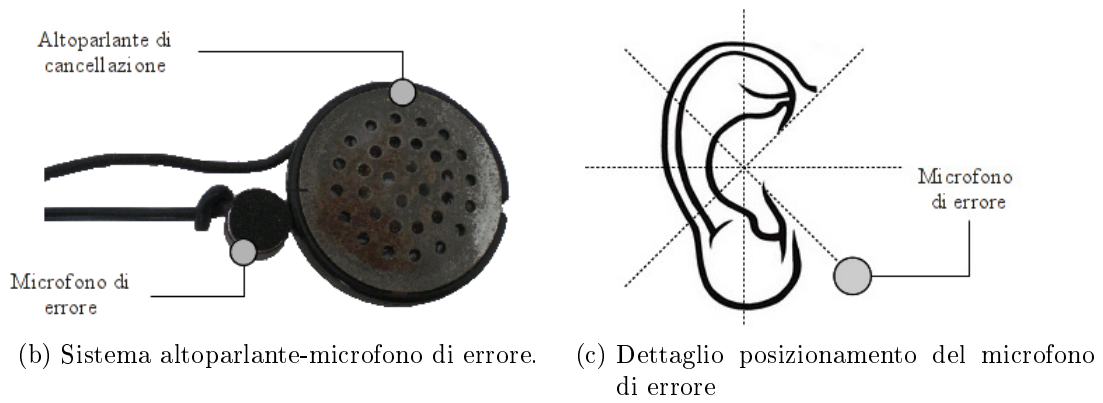


Figura 3.2.: Sensori e loro posizionamento

3.3. Applicazione dell'algoritmo FXLMS

Dopo aver ottenuto i dati necessari, si è provveduto ad eseguire delle simulazioni tramite MATLAB, applicando l'algoritmo FXLMS esposto nella sezione 2.2.

Al fine di verificare la correttezza dell'algoritmo FXLMS e di valutarne il comportamento all'aumentare del realismo, sono stati simulati tre casi distinti che si differenziano in base alla quantità di dati realistici impiegati nella simulazione.

Caso 1: il primo caso consiste nell'impiego del segnale $x(k)$ registrato dal microfono primario e nell'uso di un modello di $\hat{P}(z)$ ottenuto attraverso un processo di stima con l'ausilio del toolbox `ident`. Questa funzione di trasferimento viene stimata a partire dai segnali registrati $x(k)$ e $d(k)$. Per semplicità si è usato come modello di $S(z)$ una funzione di trasferimento unitaria.

Caso 2: in questo secondo caso vengono usati i due segnali registrati, cioè $x(k)$ e $d(k)$. Anche qui per semplicità viene usato come modello di $S(z)$ una funzione di trasferimento unitaria.

Caso 3: questo caso è il più realistico in quanto vengono utilizzati i due segnali in ingresso ottenuti dalle registrazioni, ovvero $x(k)$ e $d(k)$. Viene inoltre stimata, attraverso il toolbox `ident`, la funzione di trasferimento $\hat{S}(z)$.

Il filtro $W(z)$ è stato scelto per tutti i casi con un ordine $L = 128$. Inoltre si è scelto per semplicità di utilizzare come stima di $S(z)$ la funzione di trasferimento secondaria stessa, ovvero si è ipotizzato di essere nella condizione in cui $\hat{S}(z) = S(z)$.

Tutti e tre i casi verranno illustrati in maniera più dettagliata in seguito.

3.3.1. Primo caso

Attraverso l'analisi dei segnali $x(k)$ e $d(k)$, che rappresentano rispettivamente il rumore primario e il segnale di riferimento, si è provveduto ad ottenere una stima $\hat{P}(z)$ della funzione di trasferimento primaria $P(z)$. Si è fatto poi passare il segnale $x(k)$ attraverso tale funzione di trasferimento ottenendo così un segnale di riferimento $\hat{d}(k)$ che risulta essere ben correlato al segnale $x(k)$.

Come già detto al fine di semplificare il modello e di verificare l'efficacia dell'algoritmo si è scelto di utilizzare come funzione di trasferimento secondaria

$S(z) = 1$. Lo schema a blocchi che esemplifica questo caso è mostrato in figura 3.3.

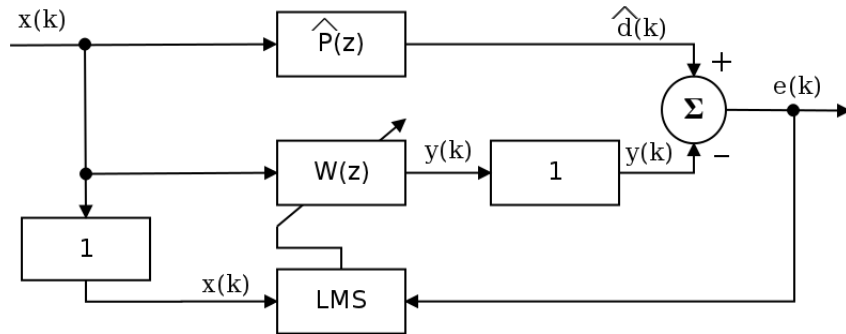
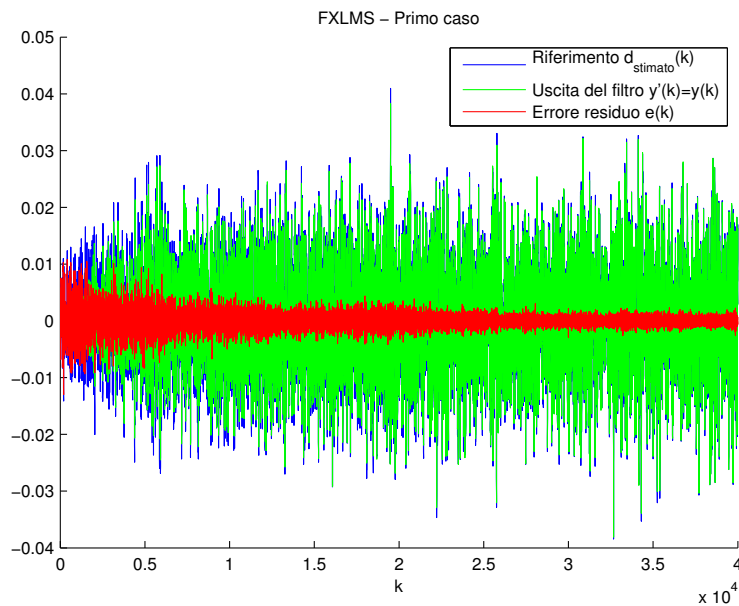


Figura 3.3.: FXLMS applicato ai caschi per motoveicoli: primo caso

3.3.1.1. Risultato della simulazione

Il risultato della simulazione è visibile in figura 3.4.



	$d(k)$	$e(k)$
Varianza:	$0.156e-3$	$2.583e-6$
Attenuazione:	41 dB	

Figura 3.4.: Risultato della simulazione: primo caso

Si può notare che il funzionamento dell' algoritmo è corretto: infatti si vede che il segnale di errore residuo $e(k)$ decresce al crescere di k .

L'attenuazione risulta di circa 41 dB: questa buona attenuazione è dovuta anche al fatto che i segnali $x(k)$ e $d(k)$ risultano ben correlati tra loro.

In altre parole il segnale $d(k)$ può venire ben approssimato dal segnale $y'(k)$ calcolato dall'algoritmo in quanto si hanno "sufficienti informazioni" a partire dal segnale $x(k)$ in ingresso all'algoritmo.

3.3.2. Secondo caso

La differenza rispetto al caso precedente sta nel fatto di non avere più il segnale $d(k)$ derivato direttamente da $x(k)$ attraverso il passaggio per $P(z)$, ma di utilizzare due segnali registrati simultaneamente² dai due microfoni.

Lo schema a blocchi che esemplifica questo secondo caso è mostrato in figura 3.5.

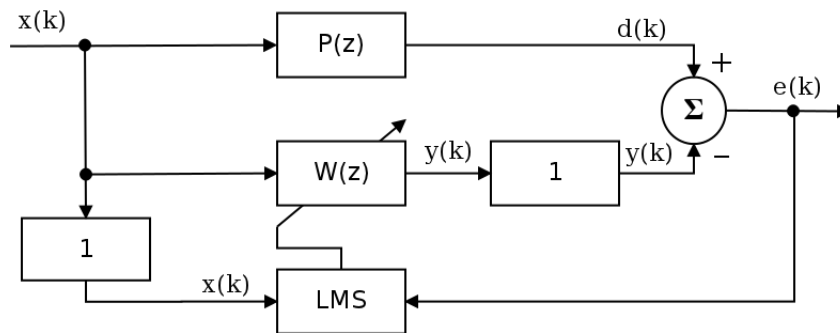


Figura 3.5.: FXLMS applicato ai caschi per motoveicoli: secondo caso

3.3.2.1. Risultato della simulazione

Il risultato della simulazione è visibile in figura 3.6.

Possiamo notare che in questa situazione, a parità di impostazioni nei parametri dell'algoritmo, il segnale di errore residuo non risulta attenuato come nel primo caso. Infatti l'attenuazione tra $d(k)$ e $e(k)$ risulta di circa 22 dB. Questo è dovuto al fatto che i segnali $x(k)$ e $d(k)$ non sono ben correlati come nel caso precedente.

²Data l'impossibilità di utilizzare un mixer per la registrazione, i due segnali $x(k)$ e $d(k)$, ottenuti rispettivamente dal microfono primario e dal microfono di errore, sono stati registrati contemporaneamente uno sul canale destro e l'altro sul canale sinistro in una registrazione stereofonica.

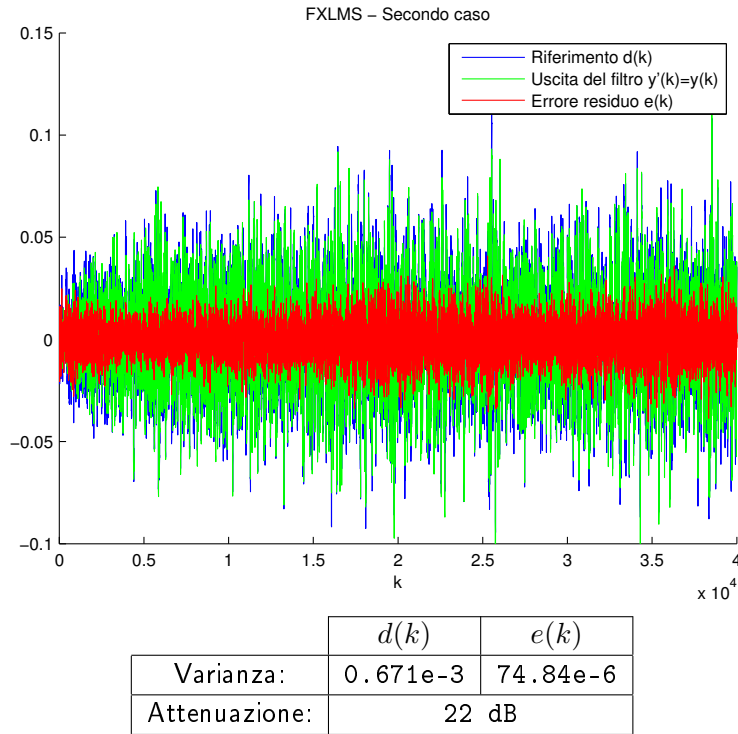


Figura 3.6.: Risultato della simulazione: secondo caso

3.3.3. Terzo caso

Come già detto questo caso è quello più realistico ed è schematizzato in figura 3.7.

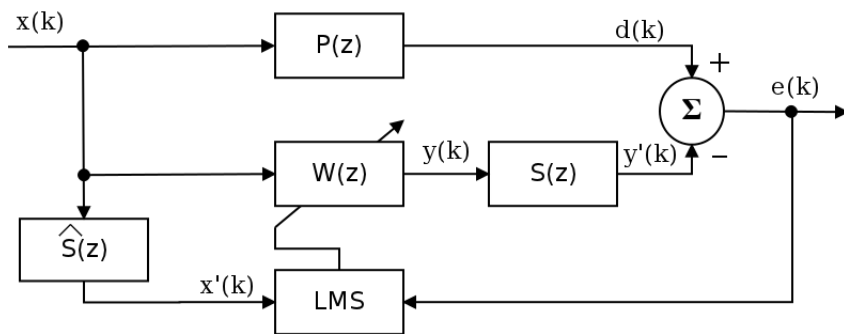


Figura 3.7.: FXLMS applicato ai caschi per motoveicoli: terzo caso

In questo caso, oltre ad utilizzare i segnali $x(k)$ e $d(k)$, viene eseguita preliminarmente una stima della funzione di trasferimento del percorso secondario $S(z)$ attraverso il metodo seguente.

3.3.3.1. Modellizzazione di $S(z)$

Come già discusso in 2.2.1.1, nelle applicazioni pratiche la funzione di trasferimento del percorso secondario $S(z)$ risulta incognita e deve pertanto essere stimata. Per fare ciò si è provveduto alla generazione tramite MATLAB di un rumore bianco il quale è stato poi inserito in ingresso all'altoparlante in esame e rilevato dal microfono di errore. Per evitare di incorrere in comportamenti non lineari dell'altoparlante esso è stato sollecitato, con un rumore bianco, solo all'interno del range di frequenze di interesse, ovvero $\Delta F = [0, 2] [kHz]^3$. Il sistema altoparlante-microfono di errore è mostrato in figura 3.2b. Questo sistema, che andrà inserito all'interno del casco, sarà quello che provvederà ad attuare la cancellazione del rumore⁴.

La funzione di trasferimento $S(z)$ ottenuta sperimentalmente è riportata tramite diagramma di Bode in figura 3.8.

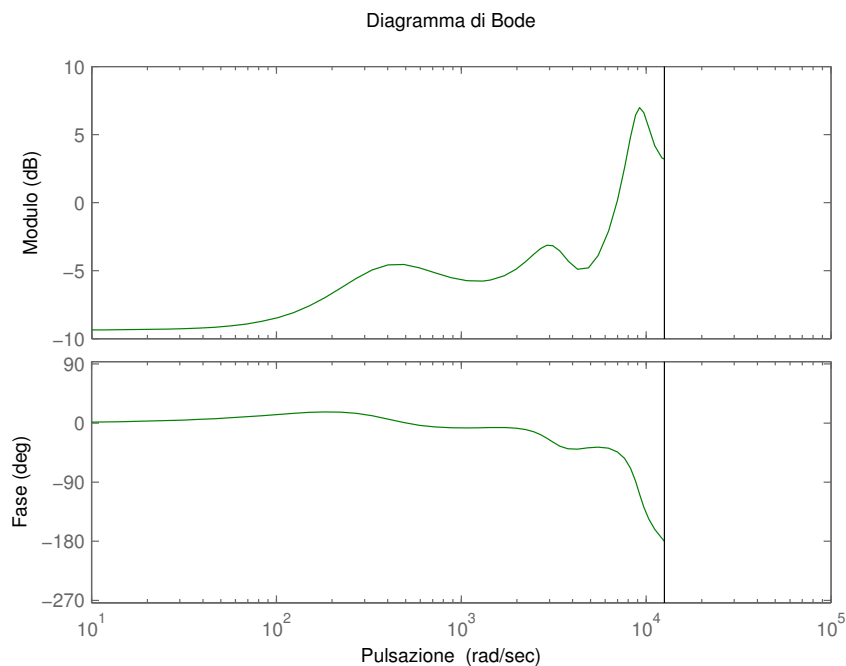


Figura 3.8.: Diagramma di Bode di $S(z)$ ottenuto sperimentalmente.

³Questo anche in considerazione del fatto che l'altoparlante usato è un altoparlante per interfono, quindi progettato per riprodurre solamente la voce umana.

⁴Come già accennato, in questa sperimentazione si è fatto uso di un solo altoparlante e di un solo microfono di errore quindi l'effettiva cancellazione avverrebbe solamente per un orecchio. Per una ipotetica applicazione commerciale si necessiterebbe di un secondo altoparlante e di un secondo microfono di errore.

Anche in questo caso si è scelto di utilizzare per la stima il toolbox `ident` ma ci si è avvalsi di un modello che esegue una stima direttamente nello spazio degli stati, impostando i vari parametri al fine di ottenere una funzione di trasferimento del sesto ordine.

3.3.3.2. Risultato della simulazione

I segnali risultanti dalla simulazione sono visibili in figura 3.9.

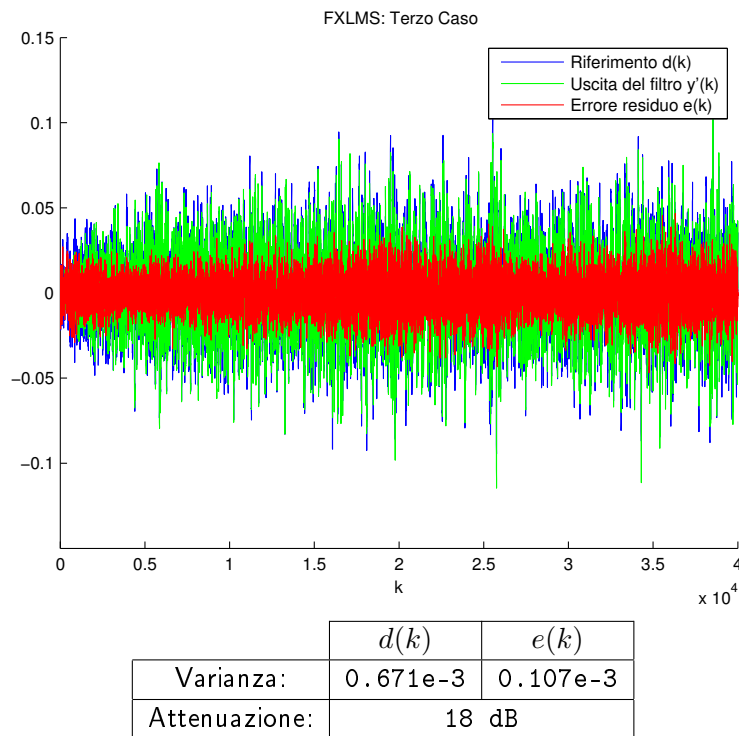


Figura 3.9.: Risultato della simulazione: terzo caso.

Anche in questo caso si nota che il filtro riesce a convergere e quindi ad attenuare il segnale residuo di errore. La limitazione risulta sempre essere dovuta al fatto che è impossibile cancellare la parte di segnale scorrelata da $x(k)$, ovvero raggiungere le prestazioni del primo caso dove i segnali $x(k)$ e $\hat{d}(k)$ sono ben correlati.

Conclusioni

I filtri adattativi risultano essere strumenti adatti ad operare in ambienti dove le conoscenze dei segnali e delle caratteristiche dei sistemi da modellizzare risultano incognite. Essi riescono infatti ad adattarsi alle situazioni in cui sono costretti ad operare.

Per le applicazioni riguardanti la cancellazione attiva del rumore è emerso che il controllo feedforward del rumore a banda larga mediante l'algoritmo FXLMS risulta essere un buon sistema di cancellazione del rumore. Come visto è necessario riuscire a modellizzare la funzione di trasferimento del percorso secondario $S(z)$ in maniera tale da "anticipare" il comportamento dell'attuatore che realizza la cancellazione. La conoscenza di tale funzione di trasferimento non deve essere necessariamente esatta in quanto il filtro adattativo si prende carico, adattandosi, di compensare anche gli effetti derivanti da questa conoscenza approssimata di $S(z)$. La stima di questa funzione di trasferimento risulta comunque essere un punto cruciale per il buon funzionamento dell'algoritmo.

Ricordando che, a causa di numerosi fattori, il controllo attivo del rumore risulta efficace per la sola cancellazione di rumore a bassa frequenza e media frequenza, la combinazione di questa tecnica con silenziatori passivi può risultare un metodo ideale per la costruzione di silenziatori operativi in un range di frequenze più ampio.

Appendice A.

Sorgenti in linguaggio Matlab

Tutti gli algoritmi presentati in questa tesi sono stati implementati utilizzando il software MATLAB. I sorgenti sono presentati di seguito.

A.1. Algoritmo LMS

```
for k=1:length(t)
    % Creo il vettore finestra su x
    if k<L
        xw=[x(k:-1:1),zeros(1,L-k)];
    else
        xw=x(k:-1:k-L+1);
    end

    %Calcolo l'uscita del filtro adattativo
    y(k)=xw*(w');

    % Calcolo il segnale d'errore
    e(k)=d(k)-y(k);

    % Aggiorno il vettore dei pesi
    w=w+mu.*xw.*e(k);
end;
```

A.2. Algoritmo NLMS

```
% Inizializzazione del vettore delle Potenze
Px_stimata_a_priori=(x(1)*x(1)') ./M;
P_min=1;
Px_stimata=0;

for k=1:length(t)
    % Creo il vettore finestra su x
    if k<L
        xw=[x(k:-1:1),zeros(1,L-k)];
    else

```

```
        xw=x(k:-1:k-L+1);
    end

    %Calcolo l'uscita del filtro adattativo
    y(k)=xw*(w');

    % Calcolo il segnale d'errore
    e(k)=d(k)-y(k);

    % Stima della potenza
    if k-M<1
        Px_stimata=Px_stimata_a_priori;
    else
        Px_stimata=Px_stimata+(x(k).^2-x(k-M).^2)/M;
    end

    % Impostazione del vincolo software su mu
    mu=alpha/(L.*max(Px_stimata, P_min));

    % Aggiorno il vettore dei pesi
    w=w+mu.*xw.*e(k);
end;
```

A.3. Algoritmo Leaky LMS

```
% Calcolo del parametro nu
nu=1-mu*gamma;

for k=1:length(t)
    % Creo il vettore finestra su x
    if k<L
        xw=[x(k:-1:1),zeros(1,L-k)];
    else
        xw=x(k:-1:k-L+1);
    end

    %Calcolo l'uscita del filtro adattativo
    y(k)=xw*(w');

    % Calcolo il segnale d'errore
    e(k)=d(k)-y(k);

    % Aggiorno il vettore dei pesi
    w=nu.*w+mu.*xw.*e(k);
end;
```

A.4. Algoritmo FXLMS

```

% Calcolo del segnale x'
x_primo=lsim(S_stimata,x);

for k=1:length(t)

    % Creo il vettore finestra su x e x'
    if k<L
        xw=[x(k:-1:1),zeros(1,L-k)];
        x_primow=[x_primo(k:-1:1),zeros(1,L-k)];
    else
        xw=x(k:-1:k-L+1);
        x_primow=x_primo(k:-1:k-L+1);
    end

    % Calcolo l'uscita del filtro adattativo
    y(k)=xw*(w');

    % Calcolo dei segnali a valle di S(z)
    if length(y)==1
        e(k)=d(k);
    else
        y_primo=lsim(S,y);
        % Calcolo il segnale di errore
        e(k)=d(k)-y_primo(k);
    end
    romani colorado 2009
    % Aggiorno il vettore dei pesi
    w=w+mu.*x_primow.*e(k);
end;

```


Appendice B.

Simbologia e nomenclatura

$x(k)$ Indica un generico segnale a tempo discreto cioè un segnale il cui dominio è un insieme numerabile inteso come insieme degli interi non negativi \mathbb{Z}_+ . Il parametro k indica il quanto temporale.

$X(z)$ Indica la trasformata Z del segnale $x(k)$, $k \in \mathbb{Z}$. Essa viene utilizzata per indicare la funzione di trasferimento di un generico sistema x .

$\mathbf{x}(k)$ Rappresenta un generico vettore \mathbf{x} all'istante k definito come $\mathbf{x}(k) = [x_0(k), x_1(k), \dots, x_L(k)]$, dove $x_i(k)$ indica il coefficiente i -esimo del vettore all'istante k .

\hat{x} Rappresenta la stima di una generica grandezza x .

$*$ Denota il prodotto di convoluzione discreto, definito come [2]:

$$[x * y](k) := \sum_{i=-\infty}^{+\infty} x(k-i)y(i) = \sum_{i=-\infty}^{+\infty} x(i)y(k-i), \quad k \in \mathbb{Z}$$

Bibliografia

- [1] S. M. Kuo, and D. R. Morgan, "*Active Noise Control Systems - Algorithms and DSP Implementations*", USA, John Wiley & Sons, Inc., 1996
- [2] G. Ricci, and M. E. Valcher, *Segnali e Sistemi*, Padova, Libreria Progetto, 2006
- [3] M. C. Lower, D. W. Hurst, A. R. Claughton, and A. Thomas, "*Sources and Levels of Noise Under Motorcyclists Helmets*", Proc. of Institute of Acoustics, vol. 16, part 2, pp. 319-325, 1994
- [4] L. Liu, S. M. Kuo, and K. P. Raghuathan, "*Active Noise Control for Motorcycle Helmet*", International Journal of Information and Communication Engineering, 6:2, pp 102-107, 2010
- [5] C. R. Fuller, A. H. von Flotow, "*Active Control of Sound and Vibration*", pp.9-19, 1995