Università Politecnica delle Marche
Scuola di Dottorato di Ricerca in Scienze dell'Ingegneria
Curriculum in Ingegneria Biomedica, Elettronica e delle Telecomunicazioni

Advanced Audio Algorithms for Enhancing Comfort in Automotive Environments

Ph.D. Dissertation of:
Francesco Faccenda

Advisor:
Prof. Francesco Piazza

Curriculum Supervisor:
Prof. Franco Chiaraluce

XIV edition - new series

Università Politecnica delle Marche
Scuola di Dottorato di Ricerca in Scienze dell'Ingegneria
Curriculum in Ingegneria Biomedica, Elettronica e delle Telecomunicazioni

Advanced Audio Algorithms for Enhancing Comfort in Automotive Environments

Ph.D. Dissertation of:
Francesco Faccenda

Advisor:
Prof. Francesco Piazza

Curriculum Supervisor:
Prof. Franco Chiaraluce

XIV edition - new series

Università Politecnica delle Marche
Scuola di Dottorato di Ricerca in Scienze dell'Ingegneria
Facoltà di Ingegneria
Via Brecce Bianche – 60131 Ancona (AN), Italy

To my family

Acknowledgments

I wish to thank Prof. Francesco Piazza for giving me the chance to live such a beautiful experience and for his availability. A special thanks also goes to ASK industries for supporting the research activity I was involved in. To Marco Vizzaccaro, my mentor and reference in last three years, whose advice and teaching have been always precious. To Luca Novarini, for all the support he gave me and the long and entertaining Skype conversations. To all the other members of ASK industries ANC-team, for the good and productive times spent together, physically and virtually. A sincere thanks also goes to Prof. Stefano Squartini, for his support and advice, without whom I would not have undertake this journey. To Emanuele Principi and Leonardo Gabrielli for their help and the whole A3Lab and Semedia crew.

Vorrei ringraziare il Prof. Francesco Piazza per avermi dato la possibilità di fare questa bellissima esperienza e per la sua disponibilità. Un ringraziamento speciale, poi, va ad ASK Industries per aver supportato l'attività di ricerca che ho portato avanti. A Marco Vizzaccaro, il mio mentore e riferimento in questi ultimi tre anni, i cui consigli ed insegnamenti sono stati sempre preziosi. A Luca Novarini, per tutto il supporto datomi e le lunghe e divertenti conversazioni su Skype. A tutti i membri dell'ANC-team in ASK Industries, per i produttivi e bei momenti passati insieme, fisicamente e virtualmente. Un sincero ringraziamento va anche a Prof. Stefano Squartini, per il suo supporto ed i suoi consigli, senza il quale non avrei intrapreso questo cammino. A Emanuele Principi e Leonardo Gabrielli per il loro aiuto e a tutto il team A3Lab e Semedia.

Ancona, January 2016

Francesco Faccenda

Abstract

Comfort within an automotive environment has become a crucial issue for car manufacturers and several efforts have been done during last decades in order to make cockpits as comfortable as possible. Acoustical comfort is an important goal in order to improve driver and passengers experience and leave them free to interact with each other. Nevertheless, in these environments, several disturbances might affect acoustic comfort, such loud noises and distances among seats.

Conversations among passengers can be assisted by suitable systems, involving the presence of microphones, amplifiers and loudspeakers. However, while on one hand these hearing aids facilitate communication, on the other acoustic couplings might arise some issues related to acoustical feedback and echo. Because of these, speech intelligibility may result ruined and, moreover, high channel gains could trigger the so called Larsen effect and drive the system to instability. Acoustic feedback and echo cancellation (respectively AFC adn AEC) methods need to be applied to keep the system stable.
In this work, a new hybrid HW/SW Test Bench, based on TMS320C6748 processor, is first proposed for testing real AFC algorithms application. AFC techniques tested are the PEM-AFROW and the Suppressor-PEM algorithms; the first one is an effective approach recently appeared in literature, while the second one has evolved from the former and exploits an additive suppressor filter included within the feedback loop in order to improve the overall system stability and increase its inherent robustness to higher gain values. Partitioned block frequency domain adaptive filter (*PB-FDAF*) paradigm has been adopted as update algorithm to keep computational complexity low. A professional sound card and a PC, where an automatic gain controller has been implemented to prevent signal clipping, complete the framework. Several experimental tests confirmed the framework suitability to operate under diverse acoustic conditions.
Subsequently, also a more realistic Dual-Channel scenario has been take into account. This means that, in contrast to Single-Channel case study, echo paths, introduced by double microphone and loudspeaker, must be considered. Voice Activity and Double Talk Detectors have been also included into algorithmic framework. Performed computer simulations in various acoustic conditions have shown the effectiveness of the approach.

Beside difficulties in communication, comfort in automotive is also considerably degraded by annoying noises, generated by engine, mechanical parts, tires on the road, or wind. Active noise control solutions can improve environment sound quality significantly, especially at low frequencies where standard passive solutions would be much bulky and expensive. This goal is achieved by playing a proper anti-noise signal through secondary sources. Narrow-band approaches have been widely discussed in the past because of their simplicity and effectiveness in the cancellation of tonal noises. In this work a novel adaptive algorithm based on narrow-band Filtered-x Least Mean Square is also proposed. Instead of adapting two taps per tone simultaneously, the suggested approach updates only one tap at a time and derives the other one from the knowledge of the adapted tap and the a-priori known amplitude of disturbing tone. This solution implicitly prevents instability conditions of standard two-taps approaches, avoiding consequent high comfort deterioration and hardware breakups. Experimental simulations have been carried out to show proposed algorithm behaviors under stress conditions, proving that its performances are comparable with those of classic narrow-band algorithm.
Moreover, adaptive filters based on Kalman control theory has been investigated as well, as an alternative for most used FxLMS-based algorithms. Kalman filter approach can easily overcome some problems of the latter ones, like slow convergence and high sensitivity to the eigenvalue spread, making it more attractive for practical applications. The main disadvantage is represented by its high computational requirements, which has prevent practical applications in the past. However, narrow-band algorithm variants significantly reduce the number of needed taps and still results competitive in a rich series of scenarios in which the noise is mainly tonal, for example generated by fans, transformers or engines. Automotive environment fit quite well in previous description and in next future it may represent a suitable scenario where apply Kalman-based ANC systems. Various possible solutions are finally presented, and their effectiveness is compared to each own FxLMS-based counterpart, with particular emphasis on multi-channel narrow-band applications, because of their great relevance in practical active noise control systems design.

Abstract

Il comfort in ambienti automotive è diventato di cruciale importanza per l'industria automobilistica e numerosi sforzi sono stati fatti nelle ultime decadi per rendere gli abitacoli i più confortevoli possibile. Il comfort acustico è un importante aspetto al fine di migliorare le impressioni di guidatori e passeggeri e lasciarli liberi di interagire fra loro. Tuttavia, in questo tipo di ambienti, numerosi disturbi possono presentarsi, come forti rumori e distanza fra i sedili.

Le conversazioni tra i passeggeri possono essere assistite da opportuni sistemi che implicano la presenza di microfoni, amplificatori ed altoparlanti. Ma se da un lato queste soluzioni, dette "hearing aids", facilitano la comunicazione, dall'altro gli accoppiamenti acustici che si instaurano possono generare problemi legati a fenomeni di feedback ed eco. A causa di questo, l'intelligibilità del parlato potrebbe essere compromessa e, soprattutto, alti guadagni di canale potrebbero portare a situazioni di instabilità dovute all'effetto Larsen. Sistemi di cancellazione di feedback ed eco acustico (rispettivamente AFC ed AEC) possono risolvere il problema in questione.
In questo lavoro, dapprima è presentato un Test Bench ibrido HW/SW basato sul processore TMS320C6748, proposto per testare applicazioni reali di algoritmi AFC. Le tecniche AFC prese in considerazione sono il PEM-AFROW ed il Suppressor-PEM; il primo è un efficace approccio apparso di recente in letteratura, mentre il secondo, evolutosi dal primo, sfrutta un ulteriori filtro soppressore incluso nel circuito di feedback per migliorare la stabilità complessiva del sistema ed incrementarne la robustezza nei confronti di alti valori di guadagno. È stato adottato il sistema di filtraggio adattativo in frequenza partizionato a blocchi (*PB-FDAF*) come sistema di aggiornamento per mantenere bassa la complessità computazionale. Completano il framework una scheda audio professionale ed un PC dove è stato implementato un sistema di controllo automatico del guadagno per evitare il clipping dei segnali. Sono stati condotti diversi test sperimentali al fine di confermare l'adeguatezza del framework di operare in numerose condizioni acustiche.
Successivamente è stato preso in considerazione anche un più realistico scenario Dual-Channel. Al contrario del caso Single-Channel, in questo sistema devono essere considerati anche gli accoppiamenti di eco acustico, dovuti ai doppi microfono ed altoparlanti. Sono stati inclusi nel framework anche un sistema di rilevamento del parlato e di double talk. Le simulazioni al computer che sono

state eseguite hanno mostrato l'efficacia dell'approccio in esame.

Oltre alle difficoltà di comunicazione, il comfort in automotive è degradato anche dai rumori fastidiosi, generati dal motore, parti meccaniche, pneumatici sulla strada od il vento. Il controllo attivo del rumore può migliorare significativamente la qualità del suono, in particolare alle basse frequenze dove le soluzioni passive standard risulterebbero troppo ingombranti e costose. L'obiettivo in questione è ottenuto suonando un opportuno segnale di anti-noise da una sorgente secondaria. Gli approcci a banda stretta sono stati ampiamente studiati in passato per la loro semplicità ed efficacia nella cancellazione dei rumori tonali. In questo lavoro si presenta anche un nuovo algoritmo adattativo basato sul narrow-band Filtered-x Least Mean Square. Invece di adattare contemporaneamente due tap per tono, il sistema proposto ne aggiorna soltanto uno e ricava l'altro dalla conoscenza del valore del tap adattato e dell'ampiezza del tono di disturbo conosciuta a priori. Questo tipo di soluzione previene implicitamente condizioni di instabilità tipiche degli approcci standard a due tap, evitando i conseguenti peggioramenti del comfort ed i breakup dell'hardware. Sono state eseguite delle simulazioni sperimentali per mostrare il funzionamento dell'algoritmo proposto in situazioni di particolare stress, dimostrando che le sue prestazioni sono comparabili con quelle dei classici algoritmi a banda stretta.

Inoltre, sono stati studiati anche i filtri adattativi basati sulla teoria dei controlli di Kalman, come alternativa ai più usati algoritmi basati sul FxLMS. L'approccio dei filtri di Kalman può facilmente superare alcuni dei problemi di quest'ultimo, come la convergenza lenta e la sensibilità alla diffusione degli autovalori, rendendolo più attraente per le applicazioni pratiche. Il principale svantaggio è rappresentato dall'alta complessità computazionale, che ne ha impedito in passato l'applicazione pratica. Tuttavia, le varianti a banda stretta dei dati algoritmi riducono significativamente il numero di tap necessari e risultano ancora competitive su una ricca serie di scenari nei quali il rumore è principalmente tonale, per esempio generato da ventole, trasformatori o motori. Gli ambienti automotive si adattano piuttosto bene alla precedente descrizione e nel prossimo futuro potrebbe essere possibile trovare applicazione ad un sistema ANC basato sul filtro di Kalman. Varie possibili soluzioni sono state, infine, presentate, e la loro efficacia è stata poi comparata con quelle delle relative controparti basate sul FxLMS, con particolare enfasi sui sistemi multicanale a banda stretta, per la loro grande rilevanza nelle applicazioni pratiche di controllo attivo del rumore.

Contents

List of Figures

List of Tables

0.1 Introduction

The cabin of a marching vehicle is continuously permeated by noise and vibrations, coming from the road, the wind and the engine itself. Audio and tactile feedbacks are continuously combined together, in a way and with an intensity strictly related to the specific environment, depending, for example, on the quality of cabin isolation, tires, windows, engine, fuel, number of people within the car and so on. These phenomena play an important role in the overall harmony of the vehicle and for this reason, driver and passengers could experience different situations from case to case, also depending on the model of the car and its brand. Possible sources for noise are manifold, and various approach to analyze and characterize them have been proposed in the literature [1]. Which component is predominant on other ones depends on speed, RPM, asphalt graininess and so no. In the following sections, from Section 0.1.1 to Section 0.1.3 the main noise sources in automotive are described, while in Section 0.1.4 active approaches for enhancing comfort in automotive are introduced.

0.1.1 Engine and mechanical components

Cockpit isolation from engine noise has improved significantly over the years. However, some issues have not been resolved yet. Users often show adverse reactions when dealing with low frequency noises, especially if they are generated by diesel engines. Unfortunately, as the wavelengths increase, classic passive isolation components get bulkier and not suitable, leaving costumers unsatisfied.
Another known issue is the so called "boom", occurring at a certain point while the engine is accelerating. It is caused by the engine transmitting excitation at its firing frequency to vehicle body panels, triggering acoustic resonances. Considering that sound quality perception is strongly related to noise characteristic changes, it appears clear that this could be perceived as a unwanted behavior. In fact, if engine loudness grows linearly with RPM, psycho-acoustic effect, for humans being, results quite enjoyable regardless of absolute SPL level; but if loudness exhibits a deviation of more than 6 dB with respect to the ideal linear trend, the focus on the noise unexpectedly increases, potentially annoying users. Same problem is presented by the noise which generates from driveline, usually consisting in few pure tones. The sound quality concern, indeed, comes from the fact that noise level varies with RMP and time, drawing people attention.

0.1.2 Road Noise

As isolation from engine improves, road noise becomes increasingly more important for overall sound quality within the cockpit. It comprehends two contributions:

- A broad-band air-rush type mainly occurring between 500 Hz and 1300 Hz.
- A tonal type, typically around 200 Hz, caused by excitement of tire acoustic cavity modes.

Its intensity strongly depend, beside on speed, also on the asphalt graininess and tires quality as well. It usually starts to be noticeable at 40 km/h but its contribution on the overall noise is maximum between 60 km/h and 100 km/h. Beyond this range, wind noise becomes predominant.

0.1.3 Wind Noise

As said in Section 0.1.2, wind noise is predominant for speed above 100 km/h. It can comprehends contributions from different aerodynamic conditions:

- Noise due to the vehicle moving at high speed through the steady air, also related to vehicle shape and its cross-sectional area.
- Noise due to turbulence through slots around doors, hood, windshield, windows etc.
- Noise due to exterior varying wind conditions, such as cross-wind.
- Beating noise due to cockpit Helmholtz resonance (around 10/20 Hz) excited by the air flow when a rear window or the sunroof are partially open.

The typical wind noise spectrum is broad-band, with a heavy predominance of low frequencies, especially for steady wind noise, which has strong components in the range from 30 Hz to 65 Hz. A gusting noise, instead, due to its impulsiveness, has content over 300 Hz as well.

0.1.4 Active Comfort Enhancing Methods

In case of very rowdy environments, understanding speeches might became hard, especially within wide cockpits, where the distances among seats are significant. Communication difficulties are often experienced as uncomfortable situations by users, especially when facing long trips with other people. Moreover, in industrial civilization, where the use of private cars is widespread, costumers are becoming more and more demanding and their expectation is to

have more and more comfortable cars as time goes by, pushing car manufactures to increasingly focus their efforts in innovative isolation systems. Passive approaches are usually cheap and effective against high frequency disturbances; nevertheless, when it comes to deal with low frequencies they become too bulky for suitable solutions. For these reason, active comfort-enhancing and speech-reinforcing methods have been extensively studied during last decades, and now that digital signal processors (DSP) are becoming powerful enough, practical applications are no longer prohibitive. These solutions are based on adaptive filters theory and attempt to predict disturbances which will be then deleted from the environment.

The thesis is organized as follow: in Chapter 1 classic adaptive filters approaches in literature are described. Chapter 2 introduces acoustic feedback and echo cancellation for communication aids in cockpits, also proposing an innovative and effective approach. Chapter 3 deals with active noise control methods, explaining two different approaches from the literature, proposing a new algorithm and showing results of a practical application in a real car. Chapter 4 concludes the thesis making few final considerations on comfort enhancing solutions described in previous chapters.

Chapter 1

Adaptive Filters for Comfort Enhancement

Adaptive filters are already widely used to cope with a variety of issues where the knowledge of the environment features is the key of the solution. Exploiting this information it is possible, for example, to predict a desired signal not directly observable or separable otherwise. Several algorithms have been carried out during past decades, descending from *gradient descent* method or control theory.
Section 1.1 introduces the classic Least Mean Square (LMS) algorithm and some of its most common variants while Section 1.2 reports first the classic Kalman filter formulation and then its application to the IRs identification problem.

1.1 Least Mean Square

Least Mean Square (LMS) [2] is a linear adaptive filtering algorithm derived from *steepest-descent*. It belongs to the family of stochastic gradient algorithms and it was developed in 1960 by Widrow and Hoff. It consists in two basic processes:

1. A filtering process, which involves:
    - A linear filter output computation in response to an input signal $x(n)$.
    - An error estimate generation comparing the linear filter output to a desired signal $d(n)$.
2. An adaptive process involving automatic filter parameters adjustment in accordance with the estimated error.

Its notoriety is due to its simplicity because it does not require neither measurements of pertinent correlation functions nor matrix inversions.

1.1.1 Classic LMS

LMS derives from *steepest-descent* approach. This latter is an iterative gradient-based algorithm which strictly follows negative gradient direction and update filter parameters as show by (1.1):

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{\mu}{2}\nabla J(n) \tag{1.1}$$

where μ is the convergence factor, also known as step-size, and $\nabla J(n)$ is the error function gradient with respect $\mathbf{w}(n)$ given by (1.2).

$$\nabla J(n) = -2\mathbf{p} + 2\mathbf{R}\mathbf{w}(n) \tag{1.2}$$

Correlation matrix R and cross-correlation vector p are respectively given by (1.3) and (1.4)

$$\mathbf{R} = E\left[\mathbf{x}(n)\mathbf{x}^T(n)\right] \tag{1.3}$$

$$\mathbf{p} = E\left[d(n)\mathbf{x}(n)\right] \tag{1.4}$$

However, in practice, prior knowledge of R and p is not available and the gradient vector $\nabla J(n)$ must be estimated. The LMS goal is to exploit instantaneous estimate of R and p:

$$\hat{\mathbf{R}} = \mathbf{x}(n)\mathbf{x}^T(n) \tag{1.5}$$

$$\hat{\mathbf{p}} = d(n)\mathbf{x}(n) \tag{1.6}$$

Substituting (1.5) and (1.6) in (1.2) in place of R and p, gradient estimate is the following:

$$\hat{\nabla} J(n) = -2d(n)\mathbf{x}(n) + 2\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{w}(n) \tag{1.7}$$

From (1.1), filter parameters update formula for LMS algorithm becomes the following:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\mathbf{x}(n)\left[d(n) - \mathbf{x}^T\mathbf{w}(n)\right] \tag{1.8}$$

Here, $\mathbf{x}^T\mathbf{w}(n)$ is the adaptive filter output and $d(n) - \mathbf{x}^T\mathbf{w}(n)$ is the error $e(n)$. Equation (1.8) could be, then, re-written in the following more familiar way:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\mathbf{x}(n)e(n) \tag{1.9}$$

At each iteration, LMS algorithm needs the most recent values of input signal $\mathbf{x}(n)$, output error $e(n)$ and filter parameters $\mathbf{w}(n)$ themselves.

1.1.2 Variable Step-Size LMS

The step-size value μ in the LMS update equation (1.9) plays a fundamental role in determine the algorithm performances. Indeed, great step-sizes assure fast convergence rates but they also imply significant errors with respect optimal solution. On the contrary, small step-sizes allow a more accurate solutions once the algorithm converged, but convergence rate results being strongly limited. Therefore, a good trade-off has to be found when classic LMS approach is adopted, in order to have a sufficient convergence rate and an acceptable error. However, several approaches have been proposed in order to dynamically change step-size value while the LMS algorithm is running. The general idea is to increase it when the algorithm is still far from optimal solution and decrease it when a steady state is being reached. Several approaches have been proposed in the literature to achieve this goal.
One of the first proposed solutions is the so called *heuristic strategy*. It consists in monitoring the instantaneous gradient estimate $\hat{\nabla} J(n)$ and check how many consecutive sign inversions have occurred. If the sign changed M_1 consecutive times, the algorithm is close to convergence and the step-size value μ might be decreased of a predefined amount C_1 in order to improve algorithm precision. If the sign did not change in the last M_2 steps, the algorithm is relatively far from convergence and the step-size might be increased of a predefined amount C_2 to increase the convergence rate.
Another important solution is the *Mathews'* approach [3], in which the step-size is updated in order to obtain a variation proportional to the gradient of squared error with respect to step-size μ itself, as show in (1.10).

$$\mu(n) = \mu(n-1) + \frac{\rho}{2} \frac{\partial e_n^2}{\partial \mu(n-1)} = \mu(n-1) + \rho e(n) e(n-1) \frac{\mathbf{x}^T(n)\mathbf{x}(n-1)}{\| \mathbf{x}(n-1) \|^2} \quad (1.10)$$

where ρ is a positive constant used to control the step-size update, and $\| \cdot \|^2$ is the Euclidean norm.
However, as reported in [4], whatever variable step-size method is adopted, convergence requirements during LMS adaptation limit the allowed step-size values, forcing the algorithm to use a bounded step-size $\bar{\mu}$ given by (1.11).

$$\bar{\mu}(n) = \begin{cases} \mu_{max} & \text{if} \quad \mu(n) > \mu_{max} \\ \mu_{min} & \text{if} \quad \mu(n) < \mu_{min} \\ \mu(n) & \text{otherwise} \end{cases} \quad (1.11)$$

In this way, instability situations due to excessive large step-sizes are avoided, as well as unresponsive ones due to excessively small values.

1.1.3 Normalized LMS

An alternative to variable step-size LMS described in Section 1.1.2 is the Normalized LMS (NLMS). This is a variant of classic LMS that attempts to decrease algorithm sensitiveness to the amplitude scaling of input signal $x(n)$. The basic idea is to divide step-size value μ by the power of input signal $x(n)$, in order to obtain smaller step-sizes when this latter is strong and greater ones when it is weak, according to the following equation:

$$\mu = \frac{\mu'}{c + \mathbf{x}^T(n)\mathbf{x}(n)} \tag{1.12}$$

where μ' is a basic constant step-size and c is an arbitrary small constant introduced to avoid division by 0 or excessively small values when $\mathbf{x}(n)$ approaches zero. NLMS algorithm is very effective when strong changes in the input signal $x(n)$ power occur, as it might happen, for example, in the case of speech signals.

1.1.4 Leaky LMS

Leakage LMS is obtained introducing in classic LMS update equation (1.9) a "leakage" factor γ, also known as "forgetting" factor. The update equation is the following:

$$\mathbf{w}(n+1) = \gamma\mathbf{w}(n) + \mu\mathbf{x}(n)e(n) \qquad \text{with} \quad 0 < \gamma < 1 \tag{1.13}$$

In this way, the effect of sample $x(n)$ at a given time step, decreases while temporal index increases. Leakage LMS might be useful in several practical applications where there are localized errors in input signal $x(n)$.

1.1.5 Sign LMS

In some practical applications, classic LMS computation burden might result too high to be handled with available hardware. In such cases, an approximation of classic LMS update equation (1.9) could be useful. Possible solutions are the following:

1. Sign Algorithm:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\text{sign}(e(n))\mathbf{x}(n) \tag{1.14}$$

2. Clipped LMS:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\text{sign}(\mathbf{x}(n)) \tag{1.15}$$

3. Sign-Sign:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \text{sign}(e(n))\text{sign}(\mathbf{x}(n)) \tag{1.16}$$

In above equations, operator "sign" denotes the sign operation reported below:

$$\text{sign}(z) = \begin{cases} 1 & \text{if} \quad z > 0 \\ 0 & \text{if} \quad z = 0 \\ -1 & \text{if} \quad z < 0 \end{cases} \tag{1.17}$$

First two solutions allow to reduce the number of multiplications, while last one completely avoids them. However, algorithm robustness decreases because the greater the approximation, the more troublesome the algorithm convergence becomes.

1.1.6 Smoothing LMS

The smoothing LMS algorithm is used when the gradient estimate $e(n)\mathbf{x}(n)$ is noisy. Depending on what kind of smoothing is applied, there are several version of this solution.
One of the most famous is the *Averaged LMS*, where the gradient estimate is substituted with its average on N samples. The update equation formula is the following:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{g}(n) \tag{1.18}$$

where $\mathbf{g}(n)$ is given by (1.19).

$$\mathbf{g}(n) = \frac{1}{N} \sum_{j=n-N+1}^{n} e(j)\mathbf{x}(j) \tag{1.19}$$

As shown by equation (1.20), the k^{th} component of $\mathbf{g}(n)$ might be also seen as a convolution with a rectangular window, whose impulse response $h(n)$ is given by (1.21)

$$g_k(n) = \frac{1}{N} \sum_{j=n-N+1}^{n} e(j)x(j-k) = h(n) * \{e(n)x(n-k)\} \tag{1.20}$$

$$h(n) = \left[\frac{1}{N}, \frac{1}{N}, \cdots, \frac{1}{N}\right] \tag{1.21}$$

Since a rectangular window is substantially a low pass filter, the gradient average $\mathbf{g}(n)$ can be generically expressed as follows:

$$\mathbf{g}(n) = h(n) * \{e(n)\mathbf{x}(n)\} \tag{1.22}$$

with $h(n)$ an impulse response of a generic low pass filter.
Another solution for smoothing LMS is the so called *Momentum LMS*. It use a simple first order FIR to obtain a gradient estimate given by equation (1.23).

$$\mathbf{g}(n) = (1-\gamma)\mathbf{g}(n-1) + \gamma\{e(n)\mathbf{x}(n)\} \qquad \text{with} \quad 0 < \gamma < 1 \tag{1.23}$$

The gradient is practically filtered with the following response:

$$G(z) = \frac{\gamma}{1-(1-\gamma)z^{-1}} \tag{1.24}$$

Since we have:

$$\mathbf{w}(n+1) - \mathbf{w}(n) = \mu\mathbf{g}(n) \tag{1.25}$$

exploiting equation (1.23) and (1.25), the Momentum LMS updating formula is the following:

$$\begin{aligned} \mathbf{w}(n+1) &= \mathbf{w}(n) + \mu\mathbf{g}(n) = \\ &= \mathbf{w}(n) + (1-\gamma)(\mathbf{w}(n) - \mathbf{w}(n-1) + \mu\gamma e(n)\mathbf{x}(n) \\ &= \mathbf{w}(n) + \mu' e(n)\mathbf{x}(n) + \alpha\left(\mathbf{w}(n) - \mathbf{w}(n-1)\right) \end{aligned} \tag{1.26}$$

where $\mu' = \mu\gamma$ and $\alpha = (1-\gamma)$. Term $\alpha\left(\mathbf{w}(n) - \mathbf{w}(n-1)\right)$ is called *momentum term*, while remaining one is the usual LMS contribution.
These approaches' goal is to improve performances at a steady state, but obviously this could affect convergence rate.

1.2 Kalman Filter

Kalman filter refers to an efficient recursive algorithm used in adaptive filters, named after its primary developer, Rudolf Emil Kálmán. This algorithms originates from a generic system identification problem [5] and it is able to obtain an estimate of system state variables exploiting system inputs/outputs measurements, even if affected by some kind of noises and inaccuracies. In this section the classic Kalman filter theory for discrete time applications is briefly described. Section 1.2.1 describes the model to which a system should fit in order to be able to exploit the Kalman filter theory. Section 1.2.2 explains how the updating process works.

1.2.1 Kalman Filter Model

In order to use the Kalman filter properly, it is important to make sure that the problem to be solved fits its conditions. Kalman filter aims find an estimate $\hat{x}_k$ of the state $x_k \in \Re^n$ of a discrete time controlled process governed by

the linear stochastic difference equation (1.27), such that the estimated output $\hat{y}_k = H_k\hat{x}_k$ is as close as possible to the ideal output $y_k = H_k x_k$. This is achieved exploiting a measurement $z_k \in \Re^m$ of y_k, given by (1.28).

$$x_k = A_k x_{k-1} + B_k u_k + w_{k-1} \tag{1.27}$$

$$z_k = y_k + v_k = H_k x_k + v_k \tag{1.28}$$

The subscript k represents the discrete time interval. Process noise and measurement inaccuracy are represented by the random variables w_k and v_k respectively. $n \times n$ matrix A_k represents the relation between the current state x_k and the previous one x_{k-1}. $n \times l$ matrix B_k represents the relation between the (optional) control inputs $u_k \in \Re^l$ and the current state x_k. $m \times n$ matrix H_k represents the relation between the current state x_k and the current output y_k. The measure z_k is an observation of the system output y_k itself.
If the system to be controlled can be modeled by previous equations, then a Kalman filter solution can be applied. Matrices A_k, B_k and H_k elements may potentially change at each time step k, but here only matrix H_k will be assumed time-varying.

1.2.2 Updating Process

Kalman filter exploits two distinct updates to achieve its goal: a prediction update, which projects forward in time the current state and error covariance estimates, and a correction update, which represents a feedback for the system, correcting the previous estimates through the knowledge of measure z_k.
The time update consists in the following two equations:

1. State prediction.
$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \tag{1.29}$$

2. Error covariance prediction.
$$P_k^- = AP_{k-1}A^T + Q \tag{1.30}$$

$\hat{x}_k^-$ and P_k^- represent the prior estimates of the state x_k and the error covariance P_k respectively, i.e. the estimates before the measurement update correction. In (1.30), Q represents the process noise covariance and its values is meant to be tuned in most case empirically.
The correction update consists in the following three equations:

1. Kalman gain computation.

$$K_k = \frac{P_k^- H_k^T}{H_k P_k^- H_k^T + R} \tag{1.31}$$

2. State estimate update via measurement z_k.

$$\hat{x}_k = x_k^- + K_k(z_k - H_k \hat{x}_k^-) \tag{1.32}$$

3. Error covariance estimate update.

$$\hat{P}_k = (I - K_k H_k) P_k^- \tag{1.33}$$

In (1.31), R represents the measure noise covariance and, unlike the process noise covariance Q, is relatively simple to determine because it refers to the noise in the environment. Equation (1.32) compute an *a posteriori* estimate of the state x_k as a linear combination between its *a priori* estimate x_k^- and a weighted difference (called residual or innovation) between an actual noisy measure z_k and a measure's prediction $H_k \hat{x}_k^-$. If these two are equal, the *a priori* state estimate $\hat{x}_k^-$ do not need any correction; otherwise, if they mismatch, $\hat{x}_k^-$ is modified. As shown in Fig. 1.1, the only signals that the update process need from the outer system are the the input matrix H_k and the measurement z_k. The algorithm proceeds iterating between the time update and the correction update at each time step, using prior values given by (1.29) and (1.30) to obtain the final estimates given by (1.31), (1.32) and (1.33). Kalman gain K is

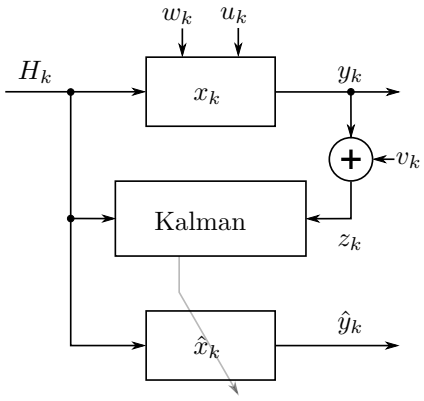


Figure 1.1: Classic Kalman filter update scheme.

an $n \times m$ matrix and the way it is computed ensure that the *a posteriori* error covariance P_k is minimized. Mathematically, (1.31) is achievable following the

steps reported below:

- Consider the *a posteriori* error equation (1.34) representing the mismatch between the actual state and *a posteriori* state $\hat{x}_k$ given by (1.32).

$$\epsilon_k = x_k - \hat{x}_k \tag{1.34}$$

- Substitute equation (1.32) in definition (1.34) and then the result in *a posteriori* error covariance equation (1.35).

$$P_k = E[\epsilon_k {\epsilon_k}^T] \tag{1.35}$$

- Derive the resulting equation with respect the Kalman gain K.

- Set the result equal to zero and solve for Kalman gain K.

From (1.31) we can observe that as the measurement error covariance R approaches zero, the Kalman gain K increases, giving more importance to the innovation $(z_k - H_k\hat{x}_k^-)$ in (1.32). This can also be interpreted as an increasing trust on measure z_k as R decreases. On the other hand, if the *a priori* error covariance P_k^- given by (1.30) decrease, Kalman gain decrease as well, which means that the measurement z_k is less important for the final state estimation.

1.2.3 Kalman Filter for IRs Identification

It is easy to fit an Impulse Response (IR) identification problem into Kalman filter theory, as shown in Fig. 1.2. Representing the IR as a Finite Impulse

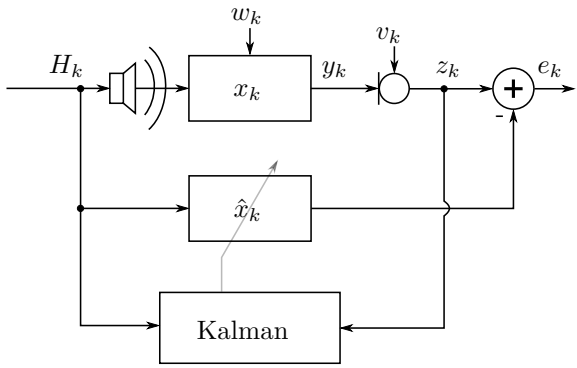


Figure 1.2: Kalman filter configuration for IR identification.

Response (FIR), the system state x_k, given by (1.36), that Kalman process

aims to estimate is represented by the FIR's taps themselves.

$$x_k = \begin{pmatrix} x_{k,1} \\ x_{k,2} \\ \vdots \\ x_{k,n} \end{pmatrix} \tag{1.36}$$

The signals y_k and z_k, instead, are the system output and its measurement respectively. Following this hints, matrix H_k, which as said before represents the relation between the current state x_k and the current output y_k, inevitably contains the system input signal r_k, often referred to as *reference*. Equation (1.37) clarifies how H_k is composed.

$$H_k = \begin{pmatrix} r_k & r_{k-1} & \cdots & r_{k-n+1} \end{pmatrix} \tag{1.37}$$

Control input signals u_k are actually optional, since in many practical applications, such as ANC systems, they are not present at all because there is no way of governing the overall environment behavior.

Chapter 2

Communicational Comfort Enhancemen Through Acoustic Feedback and Echo Cancellation

Communicational comfort might be compromised in some cases because of the noise and/or distances among seats. Hearing aids are often used in order to assist conversations, but any time an audio signal is acquired by a microphone and reproduced (once elaborated) by means of a loudspeaker within the same room environment, an acoustic coupling between the two devices arises. This means that an acoustic feedback may occur and seriously deteriorate the audio rendering performance of the overall system and limit the maximum channel gain attainable, typically expressed as maximum stable gain (MSG). Speech reinforcement techniques aim to improve the speech quality in presence of adverse acoustic conditions in order to produce a comfortable communication experience.

According to the *Nyquist Stability Criterion* [6, 7], a closed-loop system is unstable if there exist a harmonic pulse $\omega = 2\pi\left(\frac{f}{f_s}\right)$ in correspondence of which the following conditions hold:

$$\begin{cases} G(\omega,t)\,F(\omega,t) \geq 1, \\ \angle G(\omega,t)\,F(\omega,t) = n2\pi, \qquad n \in \mathbb{Z} \end{cases} \tag{2.1}$$

where $G(\omega,t)$ is the forward path transfer function and $F(\omega,t)$ the acoustic feedback path transfer function. The acoustic instability is typically exhibited as a *howling* (also known as *Larsen effect*).

Different techniques have been proposed in the literature to solve the problem [8, 9, 10]. A widely used and robust approach, belonging to the category of *gain reduction methods*, exploits notch-filter-based howling suppression (NHS) techniques. They recover from an instability situation by reducing the gain in correspondence of those frequencies where howling occurs. The cons is that, in the first place, the howling must appear in order to be detected and suppressed.

Typically, the NHS algorithms are able to detect the feedback occurrence with a significant processing latency, which acceptability in a communication scenario is questionable.
In order to suitably face the issue, solutions based on the acoustic feedback path estimate called *room modeling methods*, have been recently developed. These allows to obtain an estimate of the system feedback component which is then suitably subtracted from the acquired microphone signal. The better the estimate, the deeper the feedback cancellation will be and therefore the higher the MSG will result. Acoustic feedback cancellation (AFC) approach is based on this solution. Conceptually, the problem addressed is similar to the acoustic echo cancellation *AEC* one [11, 12], but with the substantial difference that the source signal in the *AFC* scheme is highly correlated with the signal coming from the loudspeaker (since the speaker source acts in the same acoustic environment), which results in a *biased* estimation process.
To overcome this problem, in [9, 13] an interesting technique based on the Prediction Error Method (*PEM*) is proposed. This solution, named Prediction Error Method based on Adaptive Filtering with ROW operations (PEM-AFROW), consists in decorrelating signals by means of time-varying whitening filters and then using them for the main adaptive estimation process. The main advantage is that decorrelation process does not influence the microphone signal characteristics since it takes place within the adaptive filtering circuit.
In [14] the previous approach has been further improved by means of a suppressor filter. It has been shown that the introduction of this latter can increase the performances in terms of robustness and stability, leading to an overall approach named *Suppressor-PEM*. This solution can be extended to deal with a dual-channel scenario to support a bidirectional communication. The management of two channels increases the algorithm complexity since, besides the feedback effects, echoes between channels occur as well. However, in this way the system becomes more suitable for real applications, allowing the two speakers to communicate. This idea has been previously explored in [15] and is hereby described in a more comprehensive way.
Some interesting dual-channel communication based solutions have been recently proposed in the literature. In [16] Ortega suggests an approach where only the echo effects are taken into account and additional echo suppressor filters are used to compensate the lack of AFC filters. Furthermore, in [17] and in [18], Schmidt develops an AFC system making use of a *loss control unit* which reduces the channel gains in the absence of speech and using decorrelation methods working in the closed signal loop, introducing unavoidable collateral distortion.

This chapter is organized as follows. In Section 2.1 the *PEM-AFROW* algorithm and its frequency implementation are described. Section 2.2 describes

how a suppressor filter works and the two alternative implementations suggested. In Section 2.3 a hybrid an HW/SW test-bench is proposed while Section 2.4 analyzes some experimental results obtained with aforementioned test-bench. Dual-channel issues are faced in Section 2.5 and Section 2.6 shows some simulations of dual-channels scenarios analyzing related results.

2.1 The PEM-AFROW Algorithm

The architecture of the PEM-AFROW algorithm acting on a single-channel feedback cancellation scenario is reported in Fig. 2.1 along with the notation used hereafter. In analogy to Eq. (2.1), the term $F(q)$ denotes the acoustic feedback path from the loudspeaker to the microphone, $s(t)$ and $d(t)$ the clean speech signal and the background noise respectively.

Inspired by the notation used in [19], q^{-1} denotes the unitary delay, so that $q^{-1}u(t) = u(t-1)$ and it is possible to represent a discrete time filter with finite length L as a polynomial in q:

$$F(q) = f_0 + f_1 q^{-1} + \cdots + f_{L-1} q^{-L+1} = \mathbf{f}^T \mathbf{q}. \tag{2.2}$$

The filtering operation consists in applying the polynomial to the input sequence ($F(q)u(t) = \mathbf{f}^T \mathbf{u}(t)$, $\mathbf{u}(t) = (u(t), u(t-1), \cdots, u(t-L+1))^T$). The feedback canceller $\hat{F}(q)$ estimates the feedback signal $v(t)$ (that is the result of filtering the loudspeaker signal $u(t)$ with the filter $F(q)$) and subtracts it from the microphone signal $y(t)$ before being amplified by a gain factor K and played back by the loudspeaker. The block $z^{-\Delta}$ in Fig. 2.1 models the delay always present in a real system implementation (due to A/D and D/A conversions, processing and propagation delays), that also guarantees the loudspeaker-enclosure-microphone (LEM) path identifiability, as stressed later on.

Classical direct identification techniques, like the ones used in AEC algorithms, cannot be employed to identify unknown acoustic path $F(q)$ in closed loop configuration. Indeed the basic assumption made in a typical AEC framework is that the near end signal $s(t)$ is uncorrelated with the loudspeaker signal $u(t)$, so that the acoustic path can be estimated during the echo-only period (absence of near end signal). On the contrary, in AFC problem, due to the feedback, the two signals are inevitably correlated and a direct identification of the acoustic path (i.e. performed as if the system operates in open loop) would lead to a biased feedback path estimate $\hat{F}(q)$.
Feedback path estimate is formally equivalent to a problem known in the literature as *closed-loop system identification* (CLI) [20], for which a class of solutions based on the prediction error method (PEM) can be applied, leading, under certain conditions, to an unbiased estimate of the feedback path. An in-

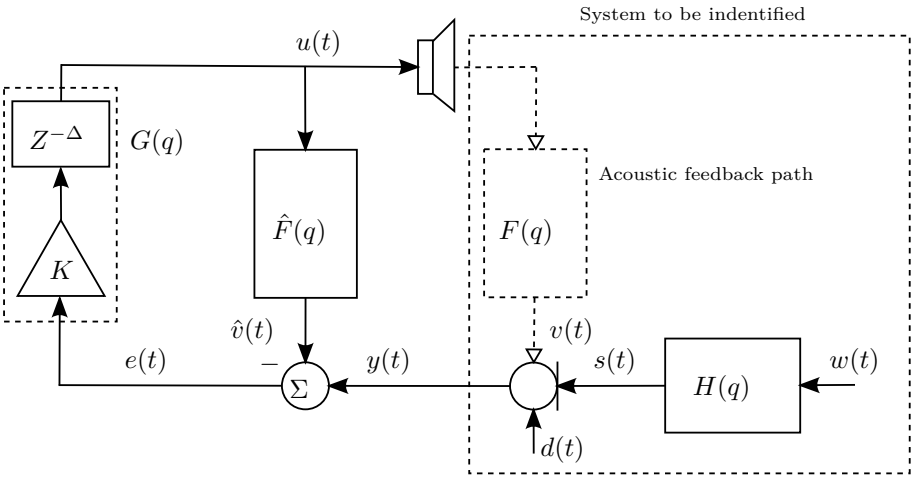


Figure 2.1: Acoustic feedback cancellation.

teresting implementation of this concept is the aforementioned PEM-AFROW algorithm. As for any PEM-based AFC algorithm, the basic assumption of the PEM-AFROW is that the near end signal, can be modeled by the expression $H(q)w(t)$, where $w(t)$ is white noise and $H(q)$ is known and inversely stable, so that the inverse filter of $H(q)$ (say $A(q) = H^{-1}(q)$) can be used to decorrelate the signals in the identification task. However, $H(q)$ is not only unknown but also time-varying, therefore it has to be estimated in an adaptive way together with $F(q)$. In [19] it is proved that if $H(q)$ is a P-order AR process, the identification of $H(q)$ and $F(q)$ can be done simultaneously, assuming the presence of a forward delay Δ bigger than the $H(q)$ filter order. PEM, applied to the system in Fig. 2.1, leads to the following cost function:

$$J_p = \frac{1}{N}\sum_{k=0}^{N-1} e_p^2(t), \quad e_p(t) = \hat{A}(q)\left[y(t) - \hat{F}(q)\,u(t)\right], \tag{2.3}$$

Equation (2.3) that must be minimized to get the optimal values for $\{\hat{A}(q), \hat{F}(q)\}$. Note that (2.3) is nonlinear in $\{\hat{A}(q), \hat{F}(q)\}$. Modelling the near end signal by a time varying AR (TVAR) model of order P, introducing a delay $\Delta > P$ in the forward path and supposing than the TVAR model is stationary over frames of 20 ms, the estimations of $A(q)$ (the inverse of the near end signal AR model) and $F(q)$ in (2.3), can be decoupled in two disjoint steps, on a frame by frame basis, as described below.
With reference to Fig. 2.1, let us model the speech source signal as a P-order

time-varying AR (TVAR) model over a L_f samples frame, as follows:

$$s(t) = a_1 s(t-1) + \cdots + a_P s(t-P) + w(t), \tag{2.4}$$

where $w(t)$ is a white noise excitation sequence. Since the feedback path is usually stationary over intervals longer than the assumed speech stationary period, we would need an approach which allows to estimate the AR model and the LEM path over frames of different length, by minimizing the cost function (2.3). Therefore, assuming the stationarity of the AR model $\mathbf{a}(t) = (1, -a_1(t), \cdots, -a_P(t))^T = \hat{\mathbf{a}}_i$, with $l = \lceil k/L_f \rceil$ over a speech frame, the PEM-AFROW is able to decouple the estimation of $\hat{\mathbf{a}}_i$ and $\hat{\mathbf{f}}(t)$ in two disjoint steps, on a frame by frame basis.
In the first step, the $\hat{\mathbf{f}}(lL_f - 1)$ estimate, obtained in the previous frame processing, is used to filter the loudspeaker signal $(t = lL_f, \cdots, (l+1)L_f - 1)$, and the result is subtracted from the correspondent values of $y(t)$, giving

$$e(t) = y(t) - \mathbf{u}^T(t)\hat{\mathbf{f}}(tL_f - 1). \tag{2.5}$$

The Levison-Durbin algorithm is then applied to the L_f frame sample values of $e(t)$ to find the linear prediction error filter $\hat{\mathbf{a}}_i$ for frame l. In the second step, the $\hat{\mathbf{a}}_l$ estimate (constant over that frame) is used and the PEM minimization problem to be solved coincides with a generic adaptive filtering problem to be applied to whitened signals. The input vector to the adaptive filtering algorithm is

$$\tilde{\mathbf{u}}^T(t) = \mathbf{a}_l^T \begin{pmatrix} \mathbf{u}^T(t) \\ \vdots \\ \mathbf{u}^T(t-P) \end{pmatrix}, \tag{2.6}$$

and the desired signal $\mathbf{a}_l^T (y(t), \cdots, y(t-P))^T$. Since $\hat{\mathbf{a}}_l$ is constant over the frame of interest, a single vector multiplication has to be performed at each iteration, and only once per frame, after the $\hat{\mathbf{a}}_l$ update, the complete vectors has to be recomputed, demanding a matrix multiplication. This makes the algorithm computationally advantageous. It must be noted that in the voiced speech case the $w(t)$ is periodic and, in order to achieve a better decorrelation, it would be preferable to implement a Long Term Prediction to model such a periodicity, in cascade to the first predictor, as effectively done in [13].
It is possible to repeat the previous two steps in an iterative way to improve the convergence; nevertheless, to keep the computational load low, one single iteration has been adopted.
The main advantage of PEM-AFROW is its pro-activity, that is the ability to cancel feedback before the howling becomes perceivable. Moreover, thanks to its remarkable identification ability, once a stable condition is achieved, it can

effectively cancel the feedback without introducing a collateral distortion in the speech signals.
To provide a real-time application for this approach, the *PB-FDAF* algorithm [2, 21] has been taken into consideration for the identification issue of $\mathbf{f}(t)$. The reasons for this choice reside in the number of advantages that this algorithm provides, i.e., fast convergence, low computational complexity and low processing latency, as confirmed by experimental results discussed in the conclusive sections.

2.1.1 Frequency-Domain Algorithm Implementation

As previously stated, a frequency-domain adaptation can lead to several advantages, including a frequency-varying step-size and a fast convergence. However, in order to work in real-time and minimize the process delay, a decrease of computational complexity is needed. This is achieved by the *PB-FDAF* algorithm which considers the impulse response to estimate as divided into M different partitions.
Let N be the total number of weights to be modeled and M be the number of blocks, so that the N-taps feedback canceler $\hat{\mathbf{f}}(t)$ can be partitioned into M segments of length $R = N/M$ each, which are then transformed to the frequency domain by means of a N_{DFT} points DFT (efficiently implemented as an FFT), with N_{DFT} equal to the smallest power of two larger than or equal to $2R$. PB-FDAF algorithm works on a M blocks queue of overlapped data and a L_f-dimensional frame of new input samples is processed at each iteration. The first step requires shifting the earlier block input vectors one position ahead into the queue, discarding the last block, while the incoming samples are converted to the frequency domain and stored on top of the queue:

$$\mathbf{U}_m(l) = \operatorname{diag}\left\{\mathcal{F}\begin{pmatrix} u((l+1)L_f - mR - N_{DFT} + 1) \\ \vdots \\ u((l+1)L_f - mR) \end{pmatrix}\right\}, \tag{2.7}$$

where $m = 0, \cdots, M-1$ and $\mathcal{F}$ is the $N_{DFT} \times N_{DFT}$ DFT matrix. The parameter $2L_f$ is the block length, which corresponds to an input/output delay of $2L_f - 1$. Since, to transform the input vector, only one DFT per block iteration is needed, a significant saving in computation can be obtained.

The output and error vectors can be expressed as:

$$\mathbf{z}_l = [\ \mathbf{0} \quad \mathbf{I}_{L_f}\ ]\mathcal{F}^{-1}\sum_{m=0}^{M-1}\mathbf{U}_m\left(l\right)\hat{\mathbf{F}}_m\left(l\right), \tag{2.8}$$

$$\mathbf{e}_l = \mathbf{y}_l - \mathbf{z}_l, \tag{2.9}$$

$$\mathbf{y}_l = \begin{bmatrix} y(lL_f+1) & \cdots & y((l+1)L_f) \end{bmatrix}, \tag{2.10}$$

$$\mathbf{E}\left(l\right) = \mathcal{F}\begin{bmatrix} \mathbf{0}_{M-L_f} \\ \mathbf{e}_l \end{bmatrix}. \tag{2.11}$$

The weight update equation is given by:

$$\hat{\mathbf{F}}_m\left(l+1\right) = \hat{\mathbf{F}}_m\left(l\right) + \mathbf{\Gamma}\left(l\right)\mathcal{F}\mathbf{g}\mathcal{F}^{-1}\mathbf{U}_m^H\left(l\right)\mathbf{E}\left(l\right), \tag{2.12}$$

where

$$\mathbf{g} = \begin{bmatrix} \mathbf{I}_R & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{N_{DFT}-R} \end{bmatrix}. \tag{2.13}$$

$\mathbf{\Gamma}\left(l\right)$ is a diagonal matrix which entries are the stepsize values $\mu_k(l)$, where k is the frequency bin. The stepsizes are normalized according to the sum of the input power $P_{u,k}\left(l\right)$ and the error power $P_{e,k}\left(l\right)$:

$$\mu_k\left(l\right) = \frac{\overline{\mu}_k}{\beta + P_{u,k}\left(l\right) + P_{e,k}\left(l\right)}, \quad k = 0, \ldots, N' - 1, \tag{2.14}$$

with

$$\begin{aligned} P_{u,k}\left(l\right) &= \textstyle\sum_{m=0}^{M-1}\|\mathbf{U}_{m,k}\left(l\right)\|^2, \\ P_{e,k}\left(l\right) &= R\|\mathbf{E}_k\left(l\right)\|^2, \end{aligned} \tag{2.15}$$

$\|\cdot\|^2$ being the squared norm operator, i.e., $\|\mathbf{x}\|^2 = \sum_{i,j} x_{i,j}^2$. This normalization of the step size with the sum of the input and error power reduces the excess error in the presence of desired signals with large power fluctuations and signal onsets. The term β is a small constant added to avoid divergence problems in silence periods.
In the current implementation, PB-FDAF algorithm has been implemented with partition length $R = L_f$.

2.2 The SUPPRESSOR-PEM Algorithm

The PEM-AFROW algorithm, especially when in stable conditions, showed to have good feedback cancellation performance without introducing distortion. Nevertheless, it is pointed out in [22] and [23] that the tracking behavior of the

PEM-AFROW is non-perfect, which may lead to an unstable system, especially when operating close to the stability threshold. In particular, changes in the acoustic feedback path (due, for example, to moving objects and/or abrupt increasing loop gain) have shown to be quite troublesome from this perspective. In presence of a dual-channel communication, such a sensitivity is relevantly stressed, as a consequence of occurrence of multiple acoustic couplings and of simultaneous adaptation of feedback and echo cancelers operating in Speech Reinforcement system.

A new approach, named *Suppressor-PEM*, employing a feedback suppression filter in cascade with the PEM-AFROW algorithm, as shown in Fig. 2.2, has been recently presented in [15]. It has been experimentally proved that such a solution is more robust with respect to the single *PEM-AFROW* algorithm. By varying the α parameter, the feedback suppressor filter switches from a prediction error filter (PEF) ($\alpha = 1$) to an infinite impulse response (IIR) adaptive oscillator ($\alpha = 0$). In PEF mode, the filter attenuates those frequencies which violates the Nyquist stability criterion, as long as any integer multiple of the inverse of the howling frequency is larger than N_D and smaller than $N_D + N_C$ samples, where N_C is the filter length and N_D is a convenient delay. On the other hand, when configured as IIR adaptive oscillator, it allows to achieve very narrow notches [17]. The N_D samples delay, usually set to 2 ms, is necessary to introduce a de-correlation in the signals, in order to avoid speech cancellation. When operating in a PEF configuration, due to the periodic or quasi-periodic nature of speech signal, the finite impulse response (FIR) filter length should not exceed 80-120 coefficients (at 16 kHz sampling rate), which roughly corresponds to a pitch period. The choice of the adaptive filtering step-size is of crucial importance, trading off between accuracy and fast convergence. Indeed, while a small value avoids speech signal suppression, a larger value guarantees faster convergence.

Since the suppressor filter length is shorter than the other adaptive filter length (i.e., the AFC filter), a time-domain based implementation for the suppressor does not exasperate too much the computational complexity of the whole system. For this reason, two algorithms have been considered for adaptive filtering in the suppressor case study: the Normalized Least Mean Squares (*NLMS*) and the frequency domain adaptive filtering (*FDAF*) [2].

2.2.1 Suppressor NLMS

Adaptive filtering algorithm adopted by this version is the *NLMS* algorithm, which operates in the time domain. In systems where the other filters work in the frequency domain (like those developed in this work) and one frame of D samples at a time is taken into account, an additional loop to consider

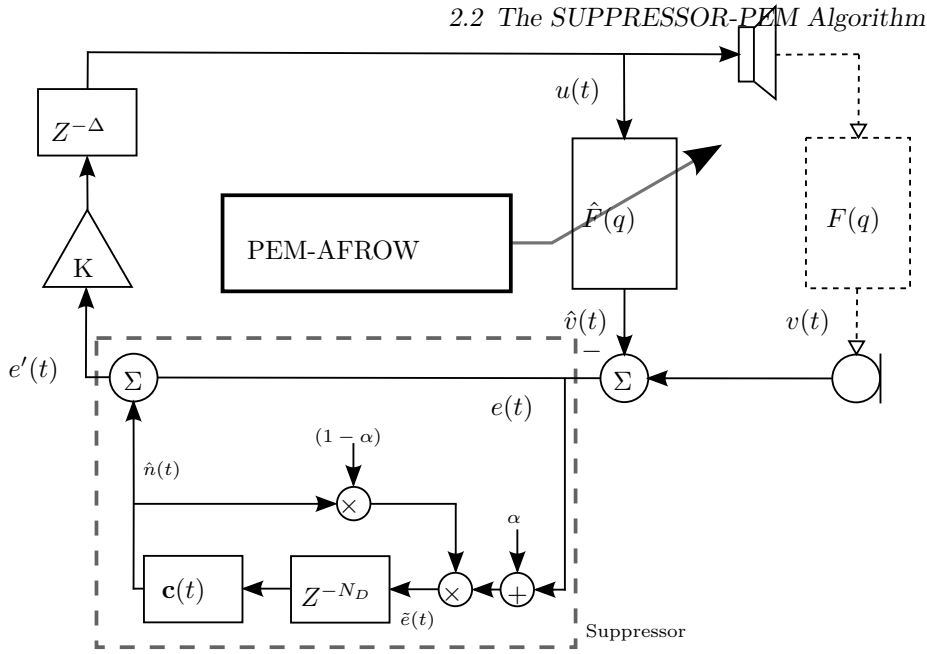


Figure 2.2: SUPPRESSOR-PEM algorithm block scheme.

one sample at a time is needed to apply this kind of suppressor filter. The behaviour of this approach within this loop is described hereafter. The signal $s(t)$ is first processed by the PEM-AFROW algorithm and the related output by the suppressor filter. The suppressor first computes the signal $\tilde{e}(t)$ as follows:

$$\tilde{e}(t) = \alpha \cdot e(t) + (1-\alpha) \cdot \hat{n}(t), \tag{2.16}$$

Subsequently the signal $\hat{n}(t)$ is computed applying the N_D samples delay to a buffer $\tilde{\mathbf{e}}(t)$ and filtering it through the suppressor filter $\mathbf{c}(t)$:

$$\hat{n}(t) = \mathbf{c}(t)^T \tilde{\mathbf{e}}(t - N_D). \tag{2.17}$$

$\hat{n}(t)$, representing the residual feedback, is then subtracted from the signal $e(t)$ to obtain $e'(t)$, which will be finally amplified and sent to the loudspeaker. The update equation for the suppressor filter using the NLMS algorithm is the following:

$$\mathbf{c}(t+1) = \mathbf{c}(t) + \mu_c(t)\tilde{\mathbf{e}}(t)e'(t), \tag{2.18}$$

with

$$e'(t) = e(t) - \hat{n}(t), \tag{2.19}$$

and

$$\mu_c(t) = \frac{\tilde{\mu}_c}{\beta + \sigma_c^2(t)}, \tag{2.20}$$

where $\mu_c(t)$ is the time-varying step-size, $\tilde{\mu}_c$ is a proper control parameter, σ_c^2 represents the variance of $\tilde{\mathbf{e}}(t)$ and β is a proper positive number which avoids divisions by either zero or too small values when $\sigma_c^2(t)$ decreases. This implementation of the Speech Reinforcement system is named *SR w/ SUPPR. NLMS*.

2.2.2 Suppressor FDAF

In this Suppressor version, adaptive filtering algorithm works in the frequency domain. Therefore, since it already considers a frame of L_f samples at a time, it does not need additional loops. The adopted algorithm is the *FDAF* (Frequency Domain Adaptive Filtering) which is substantially the same used by the AFC filters but with a single partition because of the shorter size of the suppressor filter length N_c. Indeed, it would be useful to set this latter equal to the length of an AFC filter single partition.
Just like in the time domain, the input signal is first processed by the PEM-AFROW algorithm for feedback cancellation and the AFC filter update. Afterwards a new frame l of the signal $\tilde{e}(t)$ (suppressor filter input) is computed:

$$\tilde{\mathbf{e}}(l) = \alpha \cdot \mathbf{e}(l) + (1 - \alpha) \cdot \hat{\mathbf{n}}(l). \tag{2.21}$$

Then, this is converted in frequency domain by an FFT using a number of points twice the number of samples in the time domain. To accomplish this, zero-padding in the frame queue is not recommended, since rustling may emerge in the loudspeaker signal. The adopted solution consists in keeping two previous frames in the system memory (using buffers) and properly concatenate them together with the current one to obtain a double sized frame. Two previous frames are needed, instead of just one, because last N_D samples of the current frame $\tilde{\mathbf{e}}(l)$ have to be discarded, due to the N_D samples delay to be introduced. Therefore, the samples of the previous frame $\tilde{\mathbf{e}}(l-1)$ are not enough to obtain a double sized frame, and N_D samples of the even previous one $\tilde{\mathbf{e}}(l-2)$ are needed (Fig. 2.3). Following the formalism of Section 2.1.1 and bearing in mind that in this case there is no partitioning ($M = 1$), the DFT of $\tilde{e}(t)$ is

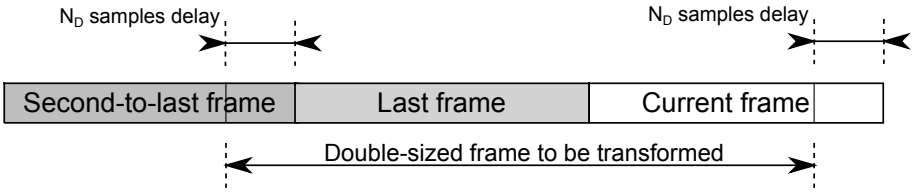


Figure 2.3: Frame concatenation before the FFT computation.

$\tilde{\mathbf{E}}(l) = \mathcal{F}(\tilde{e}(t))$. The output and error vectors can be expressed as:

$$\hat{\mathbf{n}}_l = \mathcal{F}^{-1}\tilde{\mathbf{E}}\,(l)\,\hat{\mathbf{C}}\,(l)\,, \tag{2.22}$$

$$\mathbf{e'}_l = \mathbf{e}_l - \hat{\mathbf{n}}_l, \tag{2.23}$$

$$\mathbf{e}_l = \begin{bmatrix} e(lL_f + 1) & \cdots & e((l+1)L_f) \end{bmatrix}, \tag{2.24}$$

$$\mathbf{E'}\,(l) = \mathcal{F}\mathbf{e'}_l. \tag{2.25}$$

The weight update equation is given by:

$$\mathbf{C}\,(l+1) = \mathbf{C}\,(l) + \mathbf{\Gamma}\,(l)\,\mathcal{F}\mathbf{g}\mathcal{F}^{-1}\tilde{\mathbf{E}}\,(l)\,\mathbf{E'}\,(l)\,. \tag{2.26}$$

For each frequency bin k the step size $\mu_k\,(l)$ is normalized according to the sum of the input power $P_{\tilde{E}}\,(l)$ and the error power $P_{E',}\,(l)$:

$$\mu_k\,(l) = \frac{\overline{\mu}_k}{\beta + P_{\tilde{E}}\,(l) + P_{E',}\,(l)}, \tag{2.27}$$

with

$$\begin{aligned} P_{\tilde{E}}\,(l) &= \|\tilde{\mathbf{E}}_l\,(l)\,\|^2, \\ P_{E'}\,(l) &= \|\mathbf{E'}_k\,(l)\,\|^2. \end{aligned} \tag{2.28}$$

The term β is a small constant added to avoid divergence problems in silence periods. This implementation is named *SR w/ SUPPR. FDAF*. With respect to the *SR w/ SUPPR. NLMS*, the *SR w/ SUPPR. FDAF* yields to a less accurate estimate since the FDAF algorithm is equivalent to a Block-NLMS algorithm, leading to a slightly lower effectiveness. However, the positive counterpart is a lower computational complexity leading to a faster execution, which is a primary feature for real-time applications.

2.3 The AFC Test-bench

In this section, an HW/SW test based on both a PC and an embedded system solution is described. While the embedded DSP managed one of the room-modeling based algorithm (either PEM-AFROW or Suppressor-PEM), real automotive coupling were simulated on a PC with NU-Tech.

2.3.1 OMAP-L138 eXperimenter Kit

The OMAP-L138 (Fig. 2.4) applications processor by *Texas Instruments* contains two primary CPU cores: a general-purpose ARM RISC CPU and a DSP for audio processing tasks. The former acts as the overall system controller, performing general system control tasks such as system initialization, configuration, power management, user interface, and user command implementation. The latter, which is the one exploited for the work described in Section 2.3, includes several internal memory blocks and the TI's standard *TMS320C674x megamodule* (Fig. 2.5) , which consists of the following components:

- TMS320C674x CPU
- Internal memory controllers:
  - Level 1 program memory controller (PMC)
  - Level 1 data memory controller (DMC)
  - Level 2 unified memory controller (UMC)
  - Extended memory controller (EMC)
  - Internal direct memory access (IDMA) controller
- Internal peripherals:
  - Interrupt controller (INTC)
  - Power-down controller (PDC)
  - Bandwidth manager (BWM)
- Advanced event triggering (AET)

2.3.2 NU-Tech

NU-Tech is a free software framework which allows to develop real-time DSP applications by means of a C++ based plug-in architecture. It offers a low latency interface to the sound card and can easily manage any ASIO-compatible device thanks to its remarkable capabilities of interfacing to most of the commercial USB/FireWire Audio Interfaces. The plug-in, called NUTS (NU-Tech

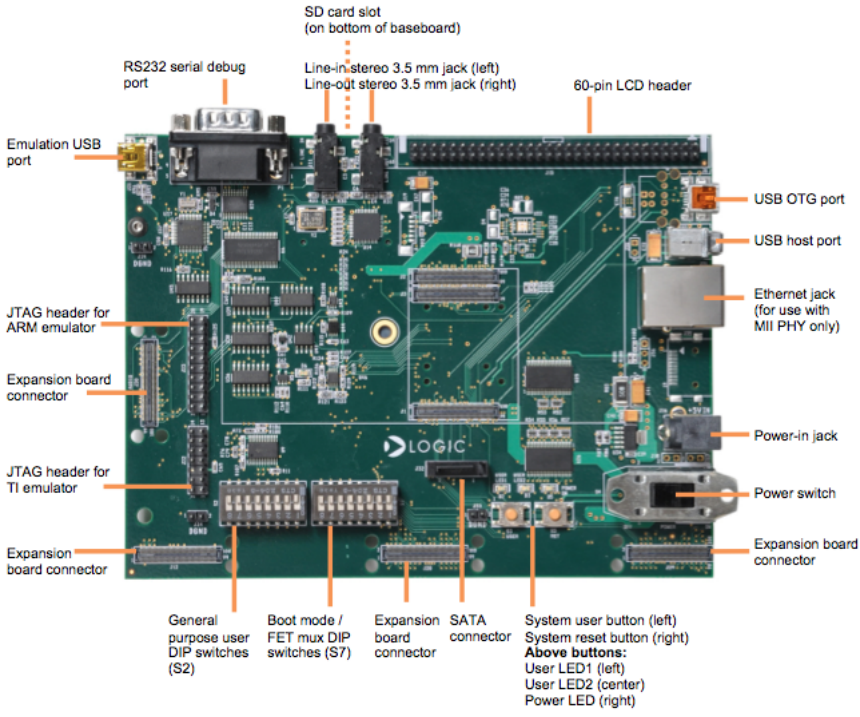


Figure 2.4: OMAP-L138.

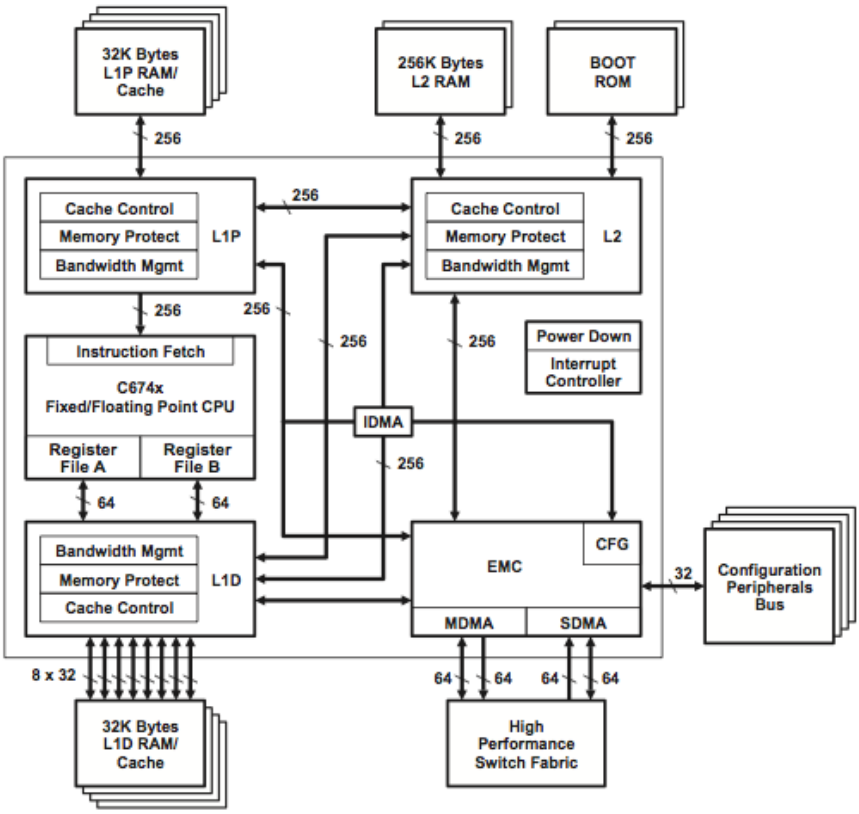


Figure 2.5: TMS320C674x megamodule.

Satellite), is a user-defined C/C++ object to be plugged-in within its panel, allowing the user to develop his own algorithms and to consider them as operating blocks.

2.3.3 Automatic Gain Controller (AGC)

The OMAP-L138 line-in audio input does not allow the capture of signal with arbitrary great power since its dynamic is limited. If a signal fed to the line-in is too strong, the signal portions which exceed a certain limit are clipped. Besides being really annoying and degrading the speech quality, this arises strong distortion problems leading to data loss and making the subsequent computation inconsistent. The adopted solution consists in normalizing the input signal amplitude to keep it within the allowed range before the OMAP-L138 captures it. Moreover, the output signal should, then, be re-amplified in order to keep the overall gain unaffected. This solution has been implemented in *NU-Tech*, before the signal is sent out and after it is captured back respectively, exploiting the larger range that the (professional) sound-card has. However, different frames and a different sample frequency are used by the OMAP-L138 and the total system delay is neither multiple of the frame time nor deterministic, but it mainly depends on the host PC CPU computational load in a given moment. For these reasons, the de-normalization procedure will hardly pick exactly the same portion of speech normalized before and if the difference between the normalization factors of two consecutive frame is great, a significantly distortion would be introduced. A straight normalization is, then, strongly not recommended and in order to minimize this problem an adaptive approach has been adopted [17]. In this solution, the normalization factor is either increased or decreased by a certain (tiny) amount depending on the current frame power P_f. In this way, even roughly knowing the total system delay, it is possible to reduce the aforementioned distortion. Given a reference power P_{ref} (found in an empiric way), the normalization factor is given by:

$$G = \begin{cases} G(1+\Delta) & \text{se} \quad P_f \cdot G^2 < P_{ref} \\ G(1-\Delta) & \text{se} \quad P_f \cdot G^2 > P_{ref} \end{cases} \tag{2.29}$$

Moreover, it has been top limited to 1, since amplifications are not allowed, and bottom limited to 10^{-6}, because of the limited precision in the elaboration that would cause an information loss. The NUTS (NU-Tech Satellite) implementing this (*AGC_forw_Schmidt*) has been defined such that it gives the possibility to set the adaptive step-size Δ, the power reference P_{ref} and the system total frame delay δ_{delay} run-time . From the latter information, the normalization NUTS just described memorizes the normalization factor used δ_{delay} frame before and share it with the de-normalization NUTS (*AGC_back*), which aims to

get the power back to the original level.
It should be noted that the greater the step-size Δ, the better the AGC copes with fast power increases. Nevertheless, this means greater difference between a normalization factor and the previous one, leading to a distortion problem. However, small step-sizes can be troublesome as well, because fast power increases could not be settled promptly, leading to the aforementioned clipping. The optimal setting is represented by the best trade-off between these two issues and it is meant to be found empirically.

2.3.4 System set-up

The algorithm described in Section 2.1 and Section 2.2 has been implemented on the OMAP-L138 eXperimenter Kit (Section 2.3.1) and tested in a real environment by means a hybrid HW/SW test bench.

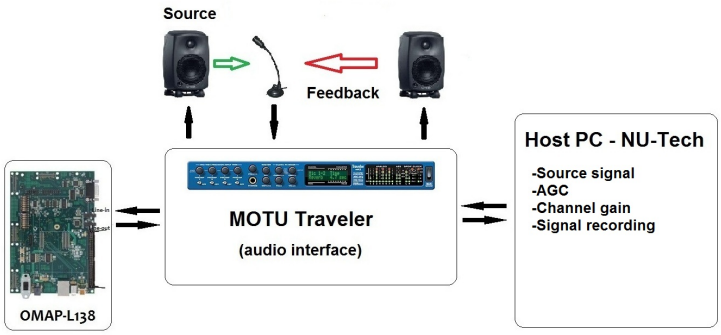


Figure 2.6: System interconnection for real environment tests.

As schematically represented in Fig. 2.6, the speech signal is first acquired by a microphone located near the source, transduced and then sent to NU-Tech platform on the host PC by means of a professional sound card, which sample it at 44.1 kHz. Here, the automatic gain controller (Section 2.3.3, NUTS *AGC_forw_Schmidt*) computes the normalization and properly attenuate the speech. The result of this operation is sent to the OMAP-L138 through the analog input and, after being re-sampled at 8 kHz, the feedback is estimated and subtracted from the signal. The compensated signal is then sent to the analog output, re-sampled at 44.1 kHz by the sound card and sent back to NU-Tech where it is de-normalized (by the *AGC_back* NUTS), amplified, recorded, and sent to the output loudspeaker. In Fig. 2.7 it is shown how the user-interface board in NU-Tech appears for this purpose.
To make the test repeatable, the signal source (i.e. a talking person) has been replaced by a NU-Tech controlled loudspeaker reproducing a speech previously

recorded at 16 kHz and 16 bits and later re-sampled at 44.1 kHz. The period of the signal is approximately 22 s (Fig. 2.8).

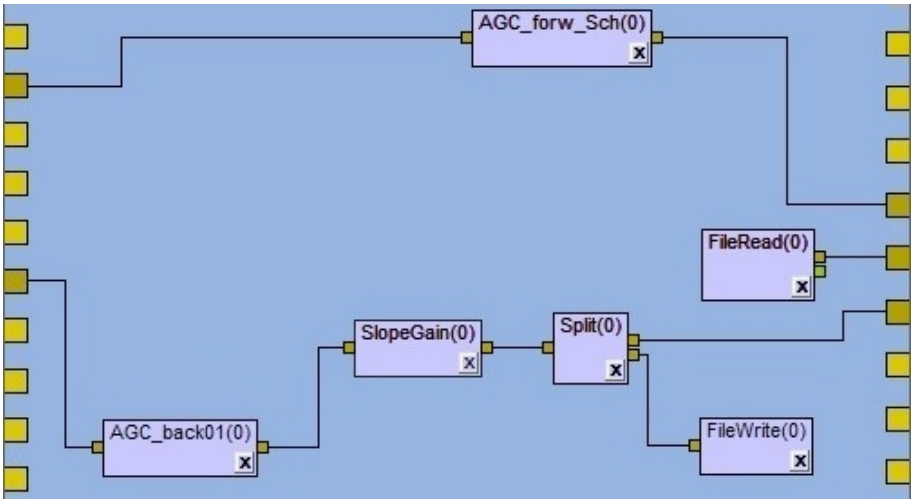


Figure 2.7: NU-Tech test-bench implementation.

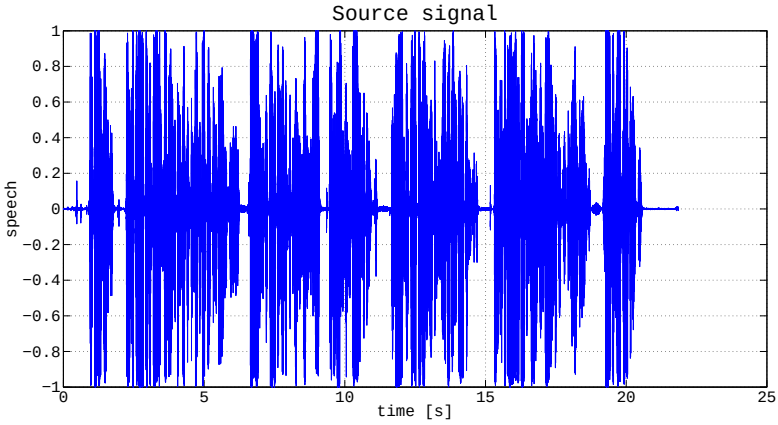


Figure 2.8: Source signal.

Fig. 2.9 shows the actual total setup implemented for the experimental tests described in Section 2.4.

The ASIO (Audio Streaming Input Output) compatible sound card used for this set-up is a *MOTU Traveler mk3* and the following setting has been used for the AGC:

- System total delay frame $\delta_{delay} = 10$

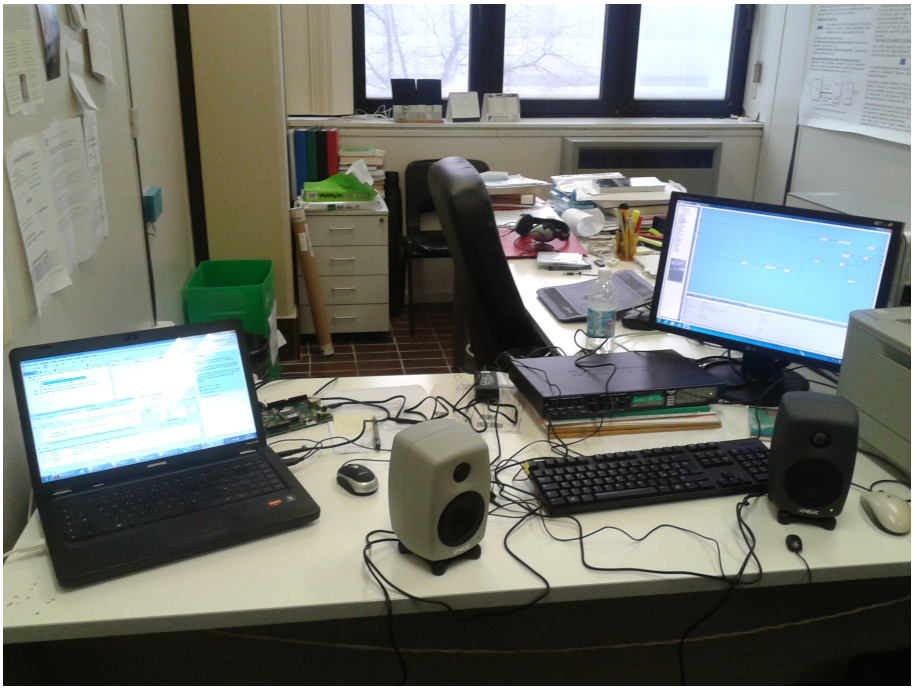


Figure 2.9: Real setup.

- Reference power $P_{ref} = 10^{14}$
- Adapting step-size $\Delta = 0.001$

The reference power P_{ref} does not refer, of course, to the absolute value, but it is meant to be related to the maximum value that can be represented by the 32 bits used in NU-TECH. Considering a normalized signal with the amplitude within $[1, -1]$, using 32 bits the previous range would be $[2^{31}, -2^{31}]$. The reference power used corresponds, then, to:

$$P_{ref} = \frac{10^{14}}{(2^{31})^2} \simeq 2.17 \cdot 10^{-5} \tag{2.30}$$

The squared denominator is due to the fact that powers are involved instead of amplitudes. The obtained value is quite low but it helps to decrease the probability that some undesired peak in the input signal would make the clipping occur.
This kind of set-up aims to test the effectiveness of the two approaches (with and without the additive suppressor filter) using real devices and without artificially simulate the loudspeaker-microphone coupling.

2.4 AFC Test-Bench Experimental Results

A linear increasing gain from 10 to 15 in a linear scale (corresponding to a range from 20 dB to 23.5 dB in a logarithmic scale) has been used. However, the values adopted are just indicative, since volume controls on the loudspeakers were present and set approximately to 1/4 of the full scale. The gain increase begins at the instant 3 s and stop at 12 s (Fig. 2.10).

The tests have been performed for different input signal-to-noise ratios (5 dB, 10 dB, 20 dB, 30 dB and 40 dB), obtained by adding a proper amount of noise to the clean speech, and repeated for a typical automotive noise (obtained from the *Noisex* database for noises[1]) and a white noise (generated in *Matlab* by the function *wgn()*). Performances of both *PEM-AFROW* approach the *Suppressor-PEM* approach have been analyzed. In order to evaluate the signal distortion introduce by the systems under test, the Itakura-Saito Measure (ISM) (2.31) and the Log-Spectral Distance (LSD) (2.32) have been evaluated on the output signals. They are both an estimate of the distance between the spectrum of a reference signal (source) and the spectrum of an elaborated signal (channel output) and they are defined as follows:

$$ISM\left(P(\omega), \hat{P}(\omega)\right) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left[\frac{P(\omega)}{\hat{P}(\omega)} - \log_{10} \frac{P(\omega)}{\hat{P}(\omega)} - 1 \right] d\omega \tag{2.31}$$

[1] *http://www.speech.cs.cmu.edu/comp.speech/Section1/Data/noisex.html*

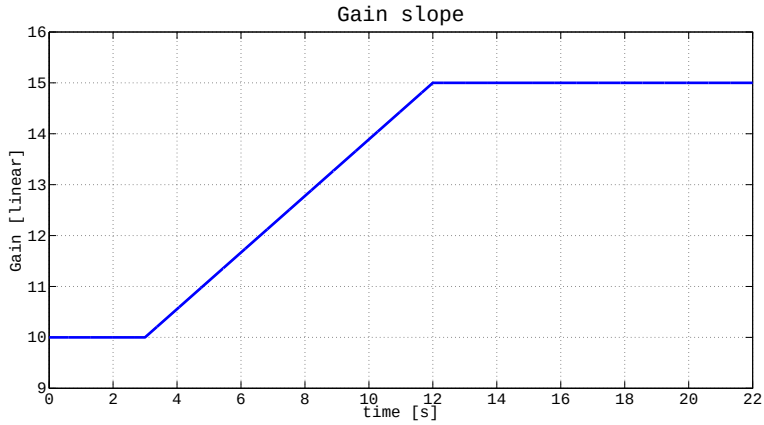


Figure 2.10: Gain slope.

$$LSD\left(P(\omega),\hat{P}(\omega)\right)=\sqrt{\frac{1}{2\pi}\int_{-\pi}^{\pi}\left[10\log_{10}\frac{P(\omega)}{\hat{P}(\omega)}\right]^{2}d\omega} \tag{2.32}$$

where $P(\omega)$ and $\hat{P}(\omega)$ are the reference signal spectrum and the elaborated signal spectrum respectively.
Since the systems used to became unstable in the end of the gain slope, the analysis of these distortion indexes have been computed on the first 2/3 of the signals, cutting them at about 15 s. Graphs in Fig. 2.11, Fig. 2.12, Fig. 2.13 and Fig. 2.14 show the results obtained in the conditions described above and highlight that in some cases, especially in automotive noise conditions, the use of an additive suppressor filter (Suppressor-PEM approach) can lead to appreciable advantages.
If white noise is added, instead, the correlation between the source signal $s(k)$ and the loudspeaker signal $u(k)$ decreases, improving the PEM-AFROW performance and making an extra suppressor filter not really useful. Nevertheless, since in real applications the noise that affects the speeches is more likely not white, the Suppressor-PEM approach could result in an interesting solution.

Tests have been executed in a office (Fig. 2.9) whose dimension in meters are about $3\times6.5\times3$. Moreover, some tests have been performed while moving the source-mic-loudspeaker relative position and the overall feeling has been that the Suppressor-PEM scheme, with respect to the PEM-AFROW one, results in a stronger and more robust approach in controlling the Larsen effect in non-static conditions. Furthermore, it was observed that in talk-through configuration (i.e. when no feedback cancellation to control the Larsen effect is applied) the system is driven almost immediately to instability, confirming

the effectiveness of the implemented algorithmic solutions.

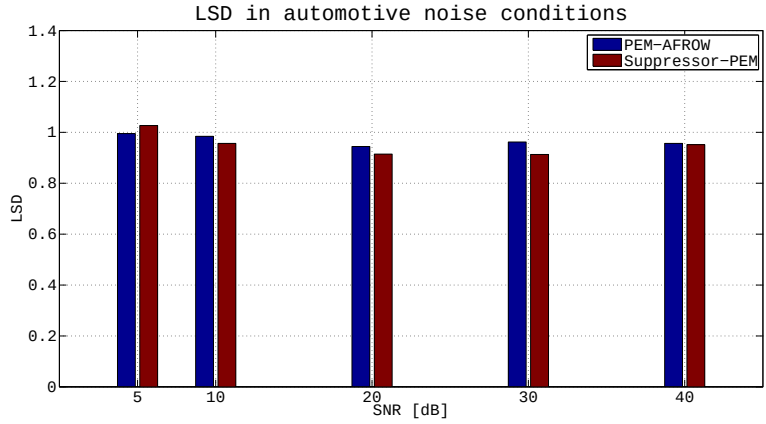


Figure 2.11: Log-Spectral Distortion in automotive noise conditions.

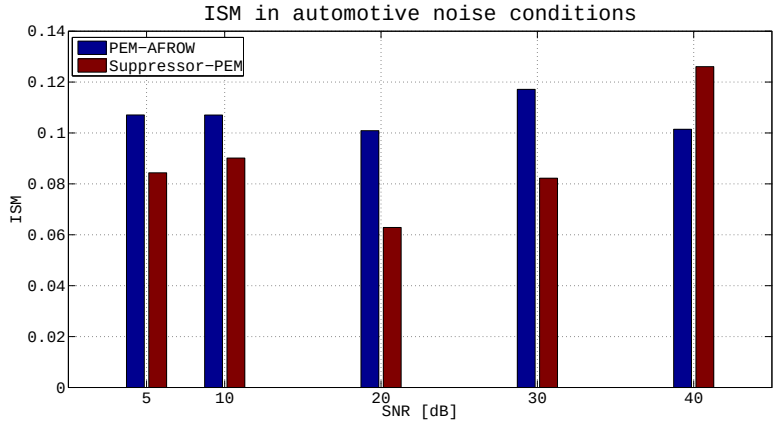


Figure 2.12: Itakura-Saito Measure in automotive noise conditions.

2.5 Dual-Channel Speech Reinforcement Systems

2.5.1 The Communication Scenario

Suppressor-PEM AFC approach described in Section 2.2 has been used for developing a dual-channel communication system which could allow two speakers to communicate in a bidirectional way. The innovation carried out in this

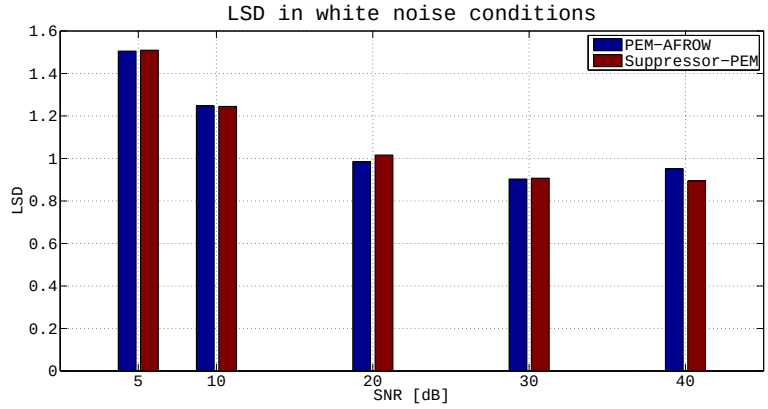


Figure 2.13: Log-Spectral Distortion in white noise conditions.

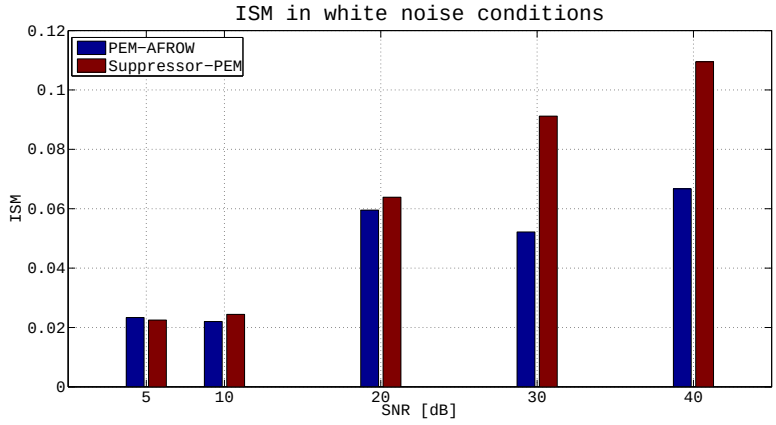


Figure 2.14: Itakura-Saito Measure in white noise conditions.

section is that the proposed system directly computes the feedback and echo components relating to the four occurring couplings, through four classic adaptive filters and two suppressor filters, leaving channel gains independent from the presence of speech.
Implemented system works in frequency domain since short execution times are preferred and in order to make real-time implementations possible. The results have been finally analyzed to confirm the improvement that the use of suppressor filters involve. Contemplating an automotive-oriented application, the speaker 1 will be referred as the one in the front side (e.g., the driver) and the speaker 2 as the one in the back side (e.g., a passenger).
Since, for a dual-channel implementation, various microphones and loudspeakers will coexist in the same environment, AFC system complexity increases with respect the single-channel scenario. The test set-up considered in this work comprises one microphone and one loudspeaker per channel. According to this, for a dual-channel scenario as discussed from now on, the number of acoustic couplings will be equal to four.
System complexity is higher than the mere summation of those of two single-channel systems because it increases quadratically with the number of channels. Indeed, there will be one feedback coupling on each channel (*rear-to-front* and *front-to-rear*) and two echo couplings between the channels (*front-to-front* and *rear-to-rear*). As a consequence the achievable *MSG* will be lower than in a single-channel solution.
For the echo compensation it is not needed (and it is rather not recommended) to use the whitened signals, since the disturbing component and the actual component are generated by different sources, so that they can be considered uncorrelated. Therefore, two AFC filters (updated by the PEM-AFROW algorithm) and two AEC filters will be needed. Besides, one suppressor filter for each channel has to be added, leading to a dual-channel system where six adaptive filters have to be managed (Fig. 2.15).

2.5.2 Algorithm Implementation

The whitening filters within the PEM-AFROW algorithm are applied to the signals $y_j(t)$ and $u_j(t)$ where $j = f, r$, in order to identify the impulse responses *rearLoud-frontMic* and *frontLoud-rearMic* respectively (Fig. 2.15). As already mentioned, for the echo couplings *frontLoud-frontMic* and *rearLoud-rearMic*, the signals whitening in their identification procedure is not needed and a standard AEC approach has been adopted. Each AEC filter is updated when a speech is detected at the far-end, and both are frozen when a *double-talk* situation occurs. This arises the need of two voice activity detectors (VAD), which are included in the algorithmic framework depicted in Fig. 2.15. They

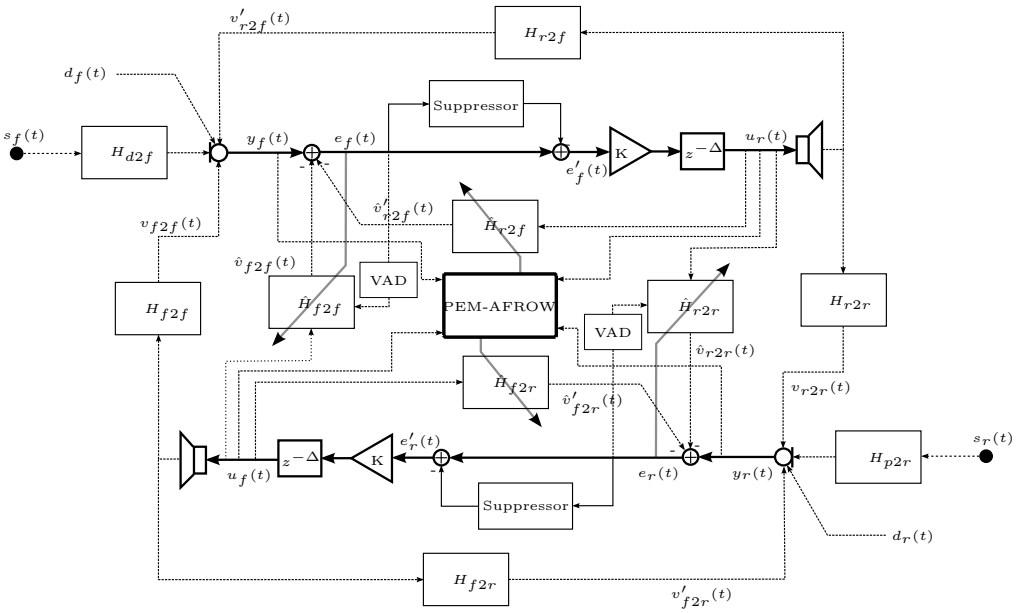


Figure 2.15: Overall Speech Reinforcement system block diagram. The f and r pedices stand for front and rear speaker position.

precisely act on the signals $e_f(t)$ and $e_r(t)$ respectively depicted in Fig. 2.15. The double talk detection is performed on the basis of the two VAD decisions: when the two VADs detect speech at the same time, then a double talk flag is set on; in all other cases is left off. The VAD algorithm used is the one proposed in [24] and it is described in Section 2.5.3. Conversely, due to the signal whitening, the AFC filters updating can always be enabled without any worry about double-talk situations.

Note that the inclusion of VADs and double-talk detection (DTD) is an element of further innovation with respect to [15]. Even though the solution adopted in this context seem to be pretty effective, as confirmed by computer simulations discussed in the following, more efforts will be devoted in the next future to implement and test other VAD and DTD algorithms, including some promising data-driven solutions recently proposed by some of the authors [25].

Each suppressor filter works independently on its channel in the way described in Section 2.2 and has an input consisting in a signal which is both feedback and echo compensated. The adaptive filtering algorithm adopted for both AFC and AEC filters is the *PB-FDAF* algorithm.

2.5.3 VAD Algorithm

The voice activity detection (VAD) algorithm used within the present framework is the one proposed in [24]. It is based on the employment of long-term speech information: this means that, in order to take a decision on a certain frame, a set of frames adjacent to it is considered.

Denoting as $x(t) = s(t) + d(t)$ a generic noisy speech signal as the sum between a clean speech signal $s(t)$ and a noise signal $d(t)$, and as $X(k,l)$ the corresponding amplitude spectrum (with k and l being the frequency bin and frame indexes respectively), the long term spectral envelope ($LTSE$) of order Q is defined as follows:

$$LTSE_Q(k,l) = \max\left\{X(k,l+j)\right\}_{j=-Q}^{j=+Q} \tag{2.33}$$

The decision rule is formulated in terms of the long term spectral divergence ($LTSD$) between the speech signal and the noise $d(t)$, defined as the divergence of the LTSE with respect to the average spectral amplitude of the noise $D(k)$ for the k-th frequency bin, with $k = 0, 1, ..., N_{DFT} - 1$. In formulas:

$$LTSD_Q(l) = 10\log_{10}\left(\frac{1}{N_{DFT}}\sum_{k=0}^{N_{DFT}-1}\frac{LTSE^2(k,l)}{D^2(k)}\right) \tag{2.34}$$

During the initialization phase, the average amplitude spectrum of the noise $D(k)$ is estimated, by assuming that the speech contribution is not present. The input signal is segmented into overlapping frames (50%) and windowed by means of the Hamming function. For the calculation of the LTSE and LTSD in equations (2.33) and (2.34) respectively, $(2Q+1)$ frames are considered. The decision is taken by comparing the LTSD value with a threshold γ, which can be calibrated on the basis of the signal energy E. In performed computer simulations a single threshold γ has been used and set equal to 16. It must be observed that the threshold choice is affected by the environmental acoustic conditions. For instance, speakers height and movements can result in diverse distances between the speaker themselves and the microphones, and therefore produce different signal levels on the two channels. In order to reduce the impact of this aspect on VAD behaviour, suitable automatic gain control strategies on both channels could be implemented on purpose, and they will be subject of future investigations.
The algorithm is designed to adapt itself to the noise variability, and therefore the spectrum $D(k)$, during the no-speech periods, is updated according to the

following:

$$D(k,l) = \begin{cases} \alpha D(k,l-1) + (1-\alpha) D_Q(k), & \text{if no-speech,} \\ D(k,l-1), & \text{otherwise,} \end{cases} \tag{2.35}$$

where α is a smoothing parameter and D_Q is the average amplitude noise spectrum calculated over Q frames given by:

$$D_Q(k) = \frac{1}{2Q+1} \sum_{j=-Q}^{Q} X(k,l+j) \tag{2.36}$$

The Q value has been set equal to 6 in performed simulations. Last but not least, an hangover mechanism is included in order to improve the speech presence detection in critical environment characterized by low SNRs.

2.6 Dual-Channel SR Simulation Results

In this section, the quality indexes used to evaluate the performance of proposed algorithm are first presented. Subsequently, a description about how input signals have been generated from a helpful database is reported. Then, adopted setup for computer simulations is discussed together with the test results obtained in different operating conditions.

2.6.1 Evaluation Criteria

In order to analyze the previously described algorithms performance and evaluate the benefits obtained by the use of suppressor filters, several simulations have been carried out in *Visual Studio* running on a PC with a *AMD Athlon II P-320 Dual-Core* processor, 2.10 GHz, 4 GB of RAM and *Windows 7* 64 bits operating system, reaching an average CPU load of approximately 12%. The evaluations have been made in an objective and repeatable way, analyzing the channels output signals and the adaptive filters impulse responses computed, as described below.

Normalized Misalignment

It is used to quantify the distance between two FIR filters, the real one and the estimated one.

$$Mis(i) = \frac{\|\mathbf{f}_i - \hat{\mathbf{f}}_i\|}{\|\mathbf{f}_i\|} \tag{2.37}$$

Subscript i denotes the frame index.

ERLE and FRLE (Echo/Feedback Return Loss Enhancement)

They provide a measure of echo and feedback attenuation, for open loop and closed loop identification respectively. Let $\langle x \rangle$ be the expectation of x, the ERLE and FRLE are defined as:

$$ERLE = 10 \log_{10} \left(\frac{1}{M} \sum_{i=0}^{M} \frac{\langle v_i^2(t) \rangle}{\langle v_i^2(t) - \hat{v}_i^2(t) \rangle} \right) \tag{2.38}$$

$$FRLE = 10 \log_{10} \left(\frac{1}{M} \sum_{i=0}^{M} \frac{\langle {v'}_i^2(t) \rangle}{\langle {v'}_i^2(t) - \hat{v'}_i^2(t) \rangle} \right) \tag{2.39}$$

$v_i(t)$ and ${v'}_i(t)$ are the echo and feedback signals at frame index i, whereas $\hat{v}_i(t)$ and $\hat{v'}_i(t)$ are the corresponding estimated signals. To obtain an overall estimate, the evaluation has been done over the whole second half of the signals (instead of each frame), while the first half has been discarded to avoid to consider transitory effects.
In single-channel applications, FRLE is often referred as ERLE. The differentiation is strictly needed when we deal with echo and feedback signals at the same time, as it occurs in the dual-channel scenario taken into account.

Itakura-Saito Measure (ISM)

It is a measure of the perceptual difference between the spectrum of a reference signal (source) and the spectrum of an elaborated signal (channel output) [26].

$$ISM\left(P(\omega), \hat{P}(\omega)\right) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left[\frac{P(\omega)}{\hat{P}(\omega)} - \log_{10} \frac{P(\omega)}{\hat{P}(\omega)} - 1 \right] d\omega \tag{2.40}$$

where $P(\omega)$ and $\hat{P}(\omega)$ are the reference signal spectrum and the elaborated signal spectrum respectively.

Log-Spectral Distance (LSD)

It is a measure of the distance between two spectra, the reference one (source) and the elaborated one (channel output).

$$LSD\left(P(\omega), \hat{P}(\omega)\right) = \sqrt{\frac{1}{2\pi} \int_{-\pi}^{\pi} \left[10 \log_{10} \frac{P(\omega)}{\hat{P}(\omega)} \right]^2 d\omega} \tag{2.41}$$

where, again, $P(\omega)$ and $\hat{P}(\omega)$ are the reference signal spectrum and the elaborated signal spectrum respectively.

Isolation

It is the measure of the isolation between two channels. Considering channel 1 (front to rear) as the reference, isolation is defined as the ratio between the

power of channel 1 output and the power of the channel 2 output when only the front passenger is talking. For the channel 2 the definition is dual.

$$I_{ch1} = 10\log_{10}\frac{\langle|u_r(t)|^2\rangle}{\langle|u_f(t)|^2\rangle} \tag{2.42}$$

$$I_{ch2} = 10\log_{10}\frac{\langle|u_f(t)|^2\rangle}{\langle|u_r(t)|^2\rangle} \tag{2.43}$$

where $u_r(n)$ and $u_f(n)$ are the output signal on the channel 1 and 2 respectively. It has to be pointed out that, in order to evaluate the achievable isolation, the algorithm needs to be close to convergence. For this reason, given the signals applied to the channels, last speech-period of each speech has been taken into account.

2.6.2 Test Setup and Simulation Results

For dual-channel systems simulation, two signals with alternated male and female speeches have been used as sources. In order to either avoid or create double-talk situations, two couples of input signals (Fig. 2.19 and Fig. 2.27 respectively) have been created. Each signal have been obtained combining seven different speech utterances recorded at 16 kHz and 16 bits from the TIMIT database [27], leading to an overall length of about 51 s in double-talk-free case. The same utterances have been combined with a partial overlap in order to make double-talk occur, thus reducing the overall length to 36 s.
The evaluation of Isolation Index values (according to the definition given above) has been performed matching last speech utterance for each channel, which occurs in time ranges [44.7–46.8] s and [48.2–50.5] s for first and second channel respectively when in double-talk-free condition.
Impulse responses used for environment simulation have been sampled at 16 kHz and they have a size of 1000 samples. In Fig. 2.16, Fig. 2.17 and Fig. 2.18 the ones relating to channel 1 are plotted. Impulse responses (*IRs*) have been acquired in the CarLab developed within the FP6 - EU funded hArtes project. CarLab stands for Car Laboratory and it is composed by a prestigious car (MERCEDES R320 CDi V6 Sport.), an advanced CIS (Car Information System) and a high-end audio system [28, 29].
Aforementioned impulse responses have been estimated by using the maximum-length sequences (MLS) method [30, 31], according to which the cross correlation between input pseudo-random noise sequence and output signal is calculated and then used to get the IR by employing the Fast Hadamard transform. These are the details of employed measurement method:

- sampling frequency = 48 kHz;

PB-FDAF		***Suppressor NLMS***		***Suppressor FDAF***	
Number of AEC/AFC filter taps	$N = 1024$	Number of taps	$N_C = 80$	Number of taps	$N_C = 128$
Partition number	$M = 8$	Suppressor filter parameter	$\alpha = 1$ (PEM mode)	Suppressor filter parameter	$\alpha = 1$ (PEM mode)
Single partition length	$R = \frac{N}{M} = 128$ (samples)	Suppressor delay	$N_D = 32$ (samples)	Suppressor delay	$N_D = 32$ (samples)
DFT points	$N_{DFT} = 2 \cdot R = 256$	Anti-divergence parameters	$\beta = 0.001$	Anti-divergence parameters	$\beta = 0.001$
Frame shift	$L_f = 128$ (samples)	Step-size	$\bar{\mu}_c = 0.001$	Step-size	$\overline{\mu}_k = 0.35 \quad \forall k$
Anti-divergence parameters	$\beta = 0.001$				
Delay	$\Delta = 128$ (samples)				
AFC step-size	$\mu_k = \begin{cases} 0.0002 & \text{if } k = 1 \\ 0.002 & \text{if } k = 2, 3, 4, N_{DFT} - 2, N_{DFT} - 1, N_{DFT} \\ 0.008 & \text{if } 4 < k < N_{DFT} - 2 \end{cases}$				
AEC step-size	$\mu_k = \begin{cases} 0.002 & \text{if } k = 1 \\ 0.02 & \text{if } k = 2, 3, 4, N_{DFT} - 2, N_{DFT} - 1, N_{DFT} \\ 0.08 & \text{if } 4 < k < N_{DFT} - 2 \end{cases}$				

Table 2.1: Algorithm parameters

- MLS order = 20;
- Spectral shape = White;
- Number of averaged MLS = 3.

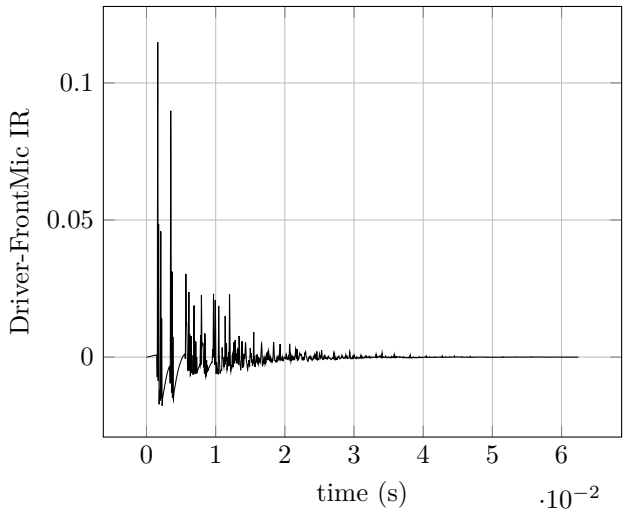


Figure 2.16: Impulse response between the front speaker and the front microphone.

Noise added to the signals is mostly typical automotive noise, that means it has an higher spectral density at low frequencies. This has been obtained from the *Noisex* database[2]. The Speech Reinforcement system has been parametrized as reported in Table 2.1.

Results of executed simulations are shown in next sections. Performance of the approach not using suppressor filters are compared with those of the approaches using the *SR w/ SUPPR. NLMS* filters and the *SR w/ SUPPR. FDAF* filters. Further simulations have been accomplished by considering different speakers and speech utterances, however no significant differences have

[2] *http://www.speech.cs.cmu.edu/comp.speech/Section1/Data /noisex.html*

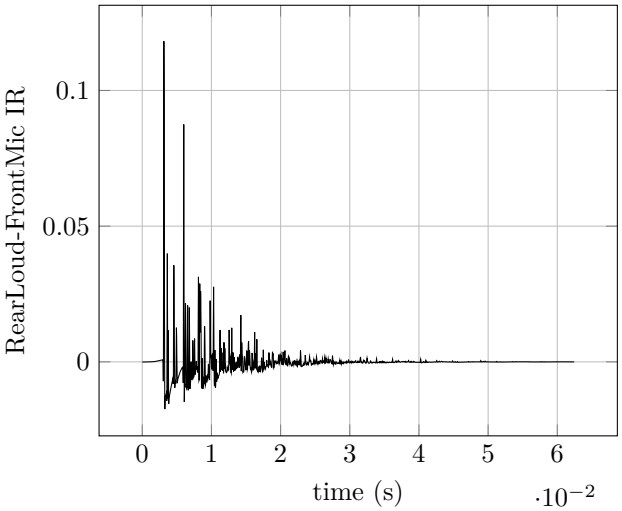


Figure 2.17: Impulse response between the rear loudspeaker and the front microphone (feedback coupling).

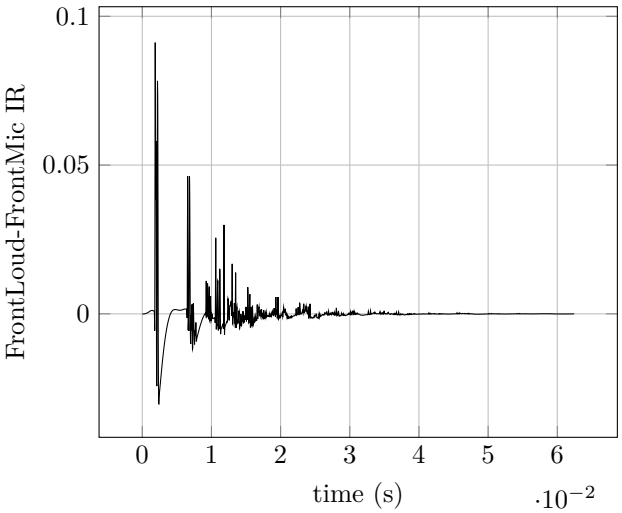


Figure 2.18: Impulse response between the front loudspeaker and the front microphone (echo coupling).

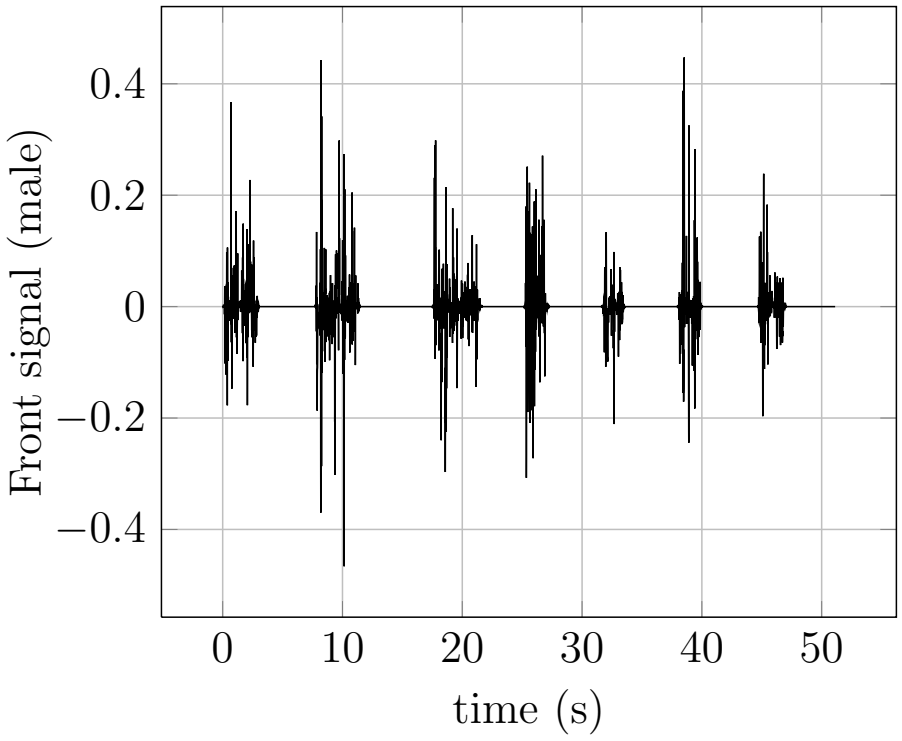
been registered in terms of framework performance. For the sake of conciseness these results have not been reported.

Variable Gain

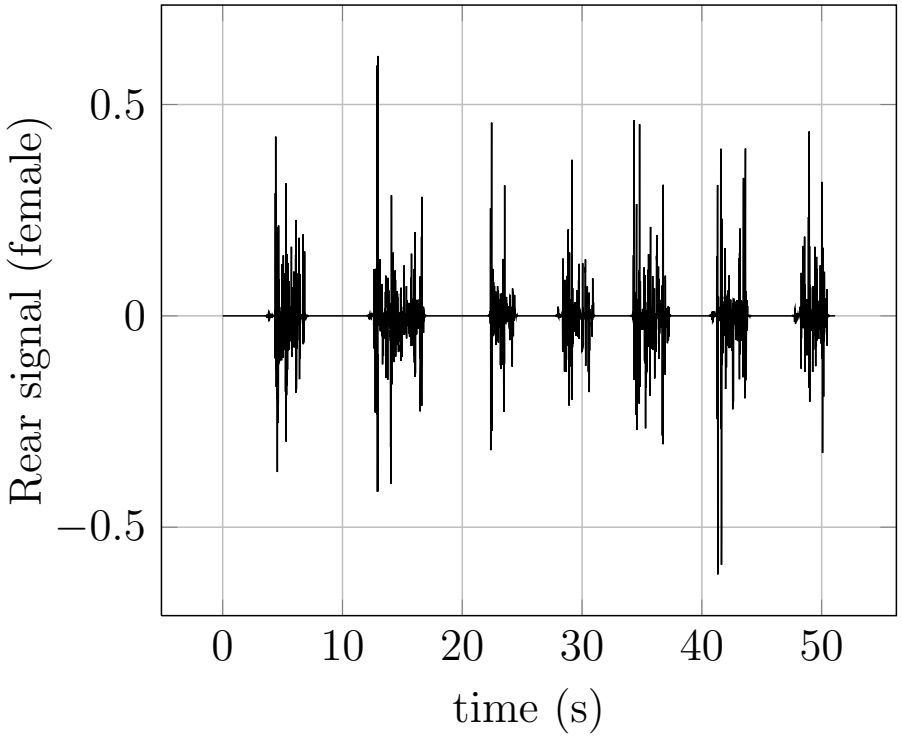
In this Section the Speech Reinforcement system performance results obtained in presence of non-overlapping speakers is analyzed by varying the gain K. Both channels have been always provided with same gain values. Starting from 0 dB, gain values were increased at each test with an increment step of 1 dB. Synthetic speech signals related to the front and rear speakers are shown in Fig. 2.19. For both channels, the input SNR is set to 40 dB.
Referring to channel 1, obtained results related to different indexes, described in Section 2.6.1, are shown in Figg. 2.22-2.26. The ones related to channel 2 are similar (as confirmed by results shown in Table 2.2, Table 2.3 and Table 2.4) and they are not reported for the sake of conciseness. White crossed pattern is assigned to the system without suppressor filters (SR BASELINE), light gray crossed pattern is for the one with Suppressor NLMS (SR w/ SUPPR. NLMS) and dark gray double crossed pattern for the one with Suppressor FDAF (SR w/ SUPPR. FDAF). It must be observed that missing bars in Figg. 2.22-2.26 means that corresponding Speech Reinforcement system is not stable and *howling* occurred. MSG value is, therefore, the highest gain value for which the system presents a bar in the plot. It is evident from those Figures that the introduction of suppressor filters within the Speech Reinforcement system allows, on one hand, to increase the system stability (from 4 dB to more than 10 dB) and, on the other, to improve the isolation between channels and decrease acoustic distortion on signals reproduced through loudspeakers. It can be also noted that the SR w/ SUPPR. NLMS version is slightly more performing then the SR w/ SUPPR. FDAF one, since an higher MSG value can be reached.
Table 2.2, Table 2.3 and Table 2.4 provide a detailed view of numerical results for all addressed quality indexes in correspondence of three gain values (3, 6, 10 dB) for both channels. As already mentioned above, same considerations can be made for both channels. Acronym *RFI* used in these Tables stands for *Recover From Instability* and means that Larsen effect actually occurred during the simulation, but the system has been able to handle it and get back into a stable condition, allowing conversation between the two speakers.

Intelligibility Tests

In order to address the intelligibility issue, the authors have first involved further objective indexes: the perceptual evaluation of speech quality (PESQ), the cepstral measure (CEP), the likelihood-ratio measure (LLR), the weighted-



(a)



(b)

Figure 2.19: Channel 1 (a) and channel 2 (b) speech signals employed to sim-

	SR baseline	SR w/ SUPPR. NLMS	SR w/ SUPPR. FDAF
		Channel 1	
Final AFC Mis [dB]	−1.41	−1.50	−1.63
Total FRLE [dB]	2.23	3.91	3.58
Final AEC Mis [dB]	−5.98	−6.10	−6.46
Total ERLE [dB]	22.58	22.00	22.57
ISM	0.01	0.01	0.01
LSD	0.49	0.54	0.57
Isolation [dB]	25.97	24.01	25.48
		Channel 2	
Final AFC Mis [dB]	−1.24	−1.31	−1.52
Total FRLE [dB]	4.00	3.37	3.40
Final AEC Mis [dB]	−5.27	−5.35	−5.34
Total ERLE [dB]	24.30	22.14	22.40
ISM	0.02	0.01	0.02
LSD	0.51	0.60	0.63
Isolation [dB]	31.06	29.74	31.39

Table 2.2: Gain=3 dB, SNR=40 dB.

	SR baseline	SR w/ SUPPR. NLMS	SR w/ SUPPR. FDAF
		Channel 1	
Final AFC Mis [dB]	Unstable	−2.07	−2.44 (RFI)
Total FRLE [dB]	Unstable	4.91	4.09 (RFI)
Final AEC Mis [dB]	Unstable	−7.23	−7.77 (RFI)
Total ERLE [dB]	Unstable	22.03	23.08 (RFI)
ISM	Unstable	0.01	0.19 (RFI)
LSD	Unstable	0.57	11.65 (RFI)
Isolation [dB]	Unstable	22.49	24.56 (RFI)
		Channel 2	
Final AFC Mis [dB]	Unstable	−2.05	−2.57 (RFI)
Total FRLE [dB]	Unstable	4.18	4.25 (RFI)
Final AEC Mis [dB]	Unstable	−6.60	−6.72 (RFI)
Total ERLE [dB]	Unstable	21.79	20.96 (RFI)
ISM	Unstable	0.01	0.21 (RFI)
LSD	Unstable	0.65	0.65 (RFI)
Isolation [dB]	Unstable	28.49	30.67 (RFI)

Table 2.3: Gain=6 dB, SNR=40 dB.

	SR baseline	SR w/ SUPPR. NLMS	SR w/ SUPPR. FDAF
		Channel 1	
Final AFC Mis [dB]	Unstable	−2.22 (RFI)	−3.19 (RFI)
Total FRLE [dB]	Unstable	3.98 (RFI)	3.63 (RFI)
Final AEC Mis [dB]	Unstable	−7.04 (RFI)	0.00 (RFI)
Total ERLE [dB]	Unstable	8.70 (RFI)	0.00 (RFI)
ISM	Unstable	1.99 (RFI)	0.11 (RFI)
LSD	Unstable	23.74 (RFI)	91.83 (RFI)
Isolation [dB]	Unstable	21.10 (RFI)	−3.53 (RFI)
		Channel 2	
Final AFC Mis [dB]	Unstable	−2.52 (RFI)	−2.10 (RFI)
Total FRLE [dB]	Unstable	4.13 (RFI)	5.78 (RFI)
Final AEC Mis [dB]	Unstable	−6.94 (RFI)	−0.20 (RFI)
Total ERLE [dB]	Unstable	17.28 (RFI)	2.04 (RFI)
ISM	Unstable	3.06 (RFI)	0.17 (RFI)
LSD	Unstable	22.00 (RFI)	82.69 (RFI)
Isolation [dB]	Unstable	21.27 (RFI)	−2.99 (RFI)

Table 2.4: Gain=10 dB, SNR=40 dB.

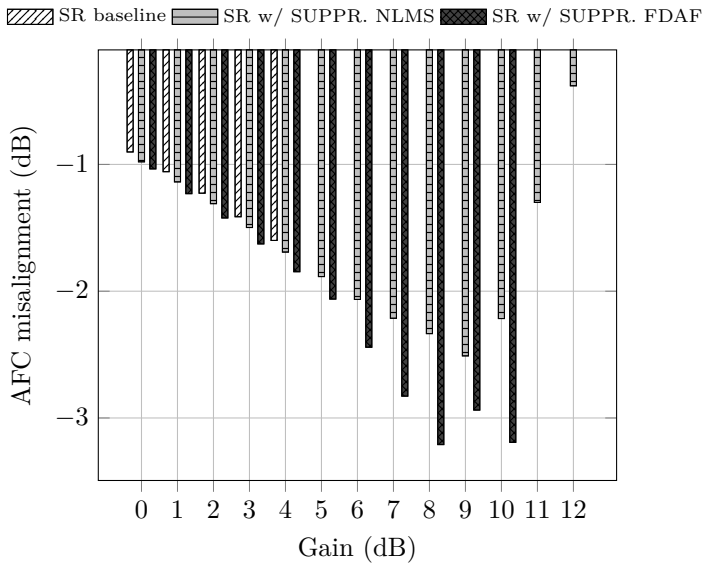


Figure 2.20: AFC misalignment in channel 1.

spectral slope metric (WSS) and the frequency weighted SNRseg (fwSNRseg) [26, 32]. As described in Loizou's work [32], their values can be used to predict intelligibility measures. They have been calculated using the output signals of Section 2.6.2 and related results confirmed the trend obtained by the measures already evaluated in that section (i.e., ISM and LSD). This allows to conclude that suppressor-based solutions show a remarkable improvement in terms of MSG and that, in this sense, the speech quality is enhanced. Therefore, taking the Loizou's work into account, speech intelligibility is also expected to increase with respect to the baseline solution.

Moving from these premises and in order to more properly quantify the intelligibility of audio files processed by the proposed framework, some subjective tests have been carried out as well. The group of people who evaluated systems performance consisted of 16 Italian subjects (8 men and 8 women) with an average age of 39.8 years (with standard deviation equal to 15.2). The algorithm evaluated are the *SR baseline* and the *SR w/ SUPPR. NLMS* with gain equal to 3 dB, 6 dB and 9 dB, with SNR=20 dB. Input signals have been composed using different Italian utterances of different speakers from *APASCI* database [33] and each signal had an overall length of about 70 s, with 30 s of speech and about 60 words. This database fitted quite well for this purpose, since its utterances are sequence of words without a real semantic meaning: this reduces the possibility for a listener to understand one or more words from the context, which introduces bias in the evaluation process. The output signals from the two algorithms under test in the 3 different operating conditions have

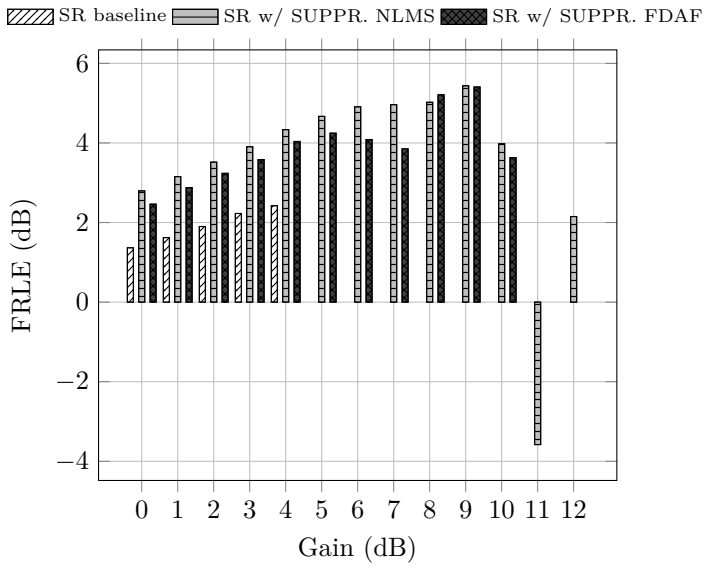


Figure 2.21: FRLE in channel 1.

been recorded and then randomly presented to the subjects. During listening, the subjects were allowed to stop the reproduction, report what they heard and then proceed with listening. Rewind was not allowed, since intelligibility at first hearing was meant to be evaluated. In order to make the evaluation faster, only output signals of first channel (where the relative speakers always start the conversation) have been submitted to the listeners. The average time required to each listener to evaluate both algorithms in the 3 different conditions was 15 minutes and 15 seconds, with a standard deviation equal to 1 minute and 54 seconds. In Table 2.5 the results averaged over all subjects are reported.

	Gain = 3 dB		**Gain = 6 dB**		**Gain = 9 dB**	
	avg	**std**	**avg**	**std**	**avg**	**std**
SR baseline	90.4%	7.5%	0.0%	0.0%	0.0%	0.0%
SR w/ SUPPR. NLMS	90.6%	8.0%	90.2%	7.8%	74.6%	9.8%

Table 2.5: Average percentage of understood words (avg) in intelligibility tests and relative standard deviation (std).

Presented results allow concluding that the system becomes completely unintelligible when it is driven to instability (e.g., the *SR baseline* at 6 dB and 9 dB), while it is almost completely intelligible when in stable condition, suggesting that the intelligibility is mainly influenced by the noise presence. The decrease of understood words in *SR w/ SUPPR. NLMS* evaluation at 9 dB is due to the initial instability that, however, the system managed to recover. Therefore, the use of suppressor filters allows to remarkably improve the sys-

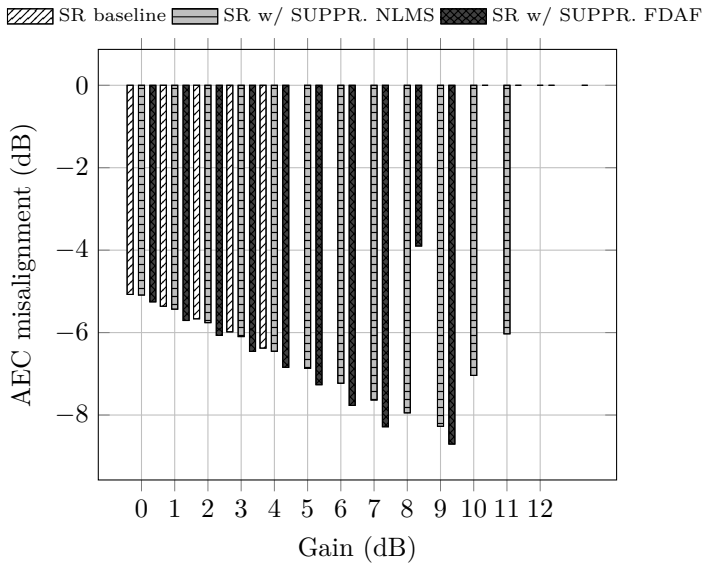


Figure 2.22: AEC misalignment in channel 1.

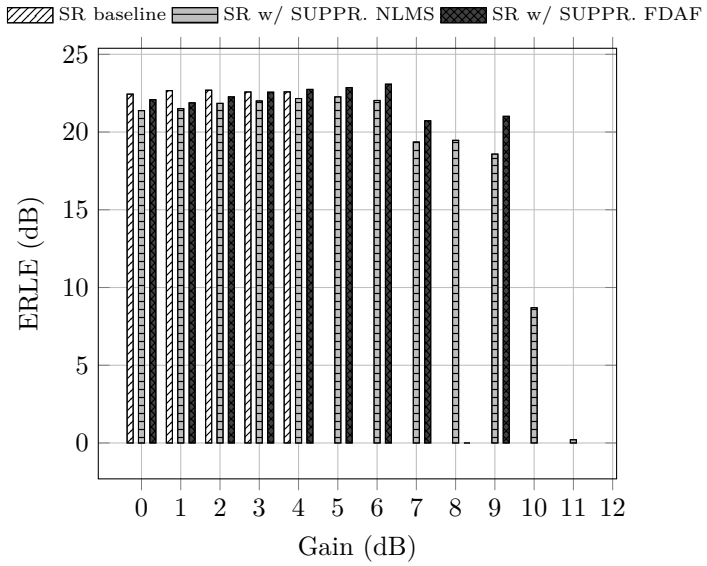


Figure 2.23: ERLE in channel 1.

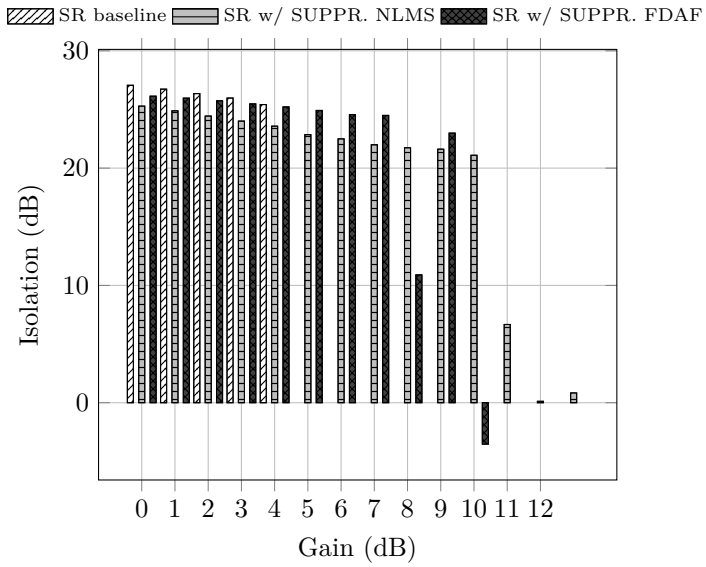


Figure 2.24: Isolation in channel 1.

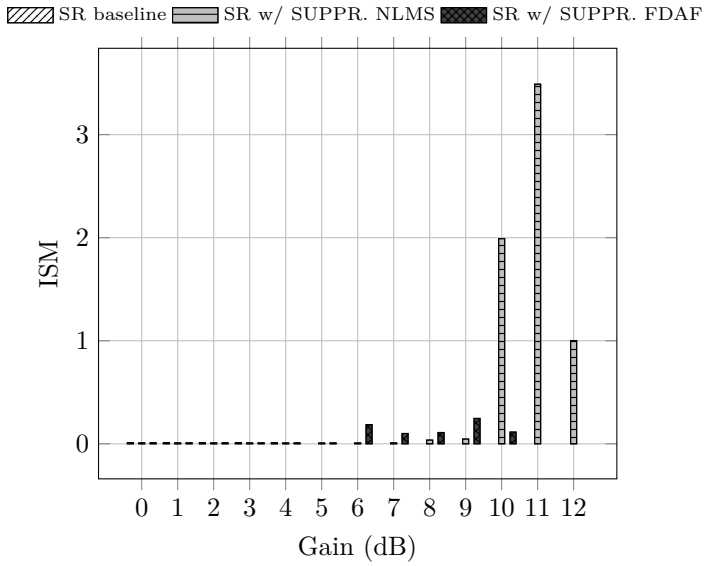


Figure 2.25: ISM in channel 1.

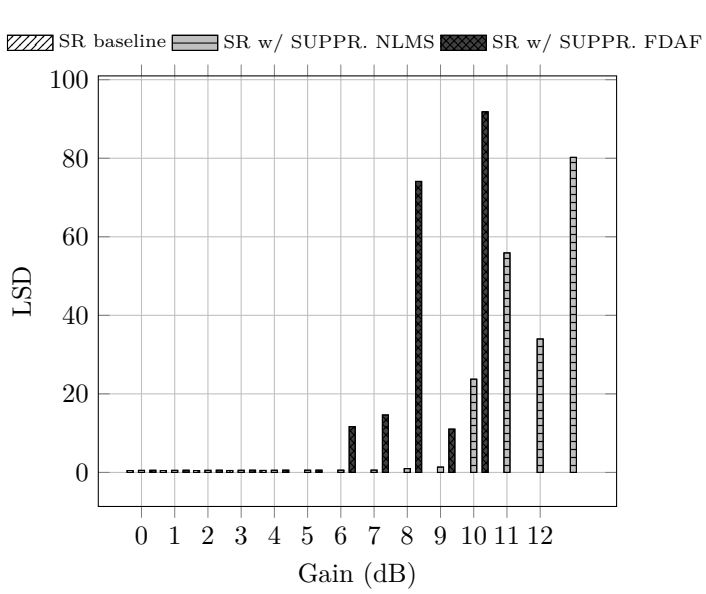


Figure 2.26: LSD in channel 1.

tem intelligibility also in presence of high gain values. Further intelligibility estimation objective criteria, discussed in [34, 35], will be taken into account in future investigations.

Evaluation With Different Speech Samples

Further simulations have been performed using different input signals couples in order to observe the behavior of the *SR w/ SUPPR. NLMS* algorithm when speakers change. As for signals used in Section 2.6.2, new ones have been generated combining seven speeches recorded at 16 kHz and 16 bits from the *TIMIT* database without overlap, leading to an overall length (including silence) of about 51 s. The amount of speech for each of them is about 20 s.
In Table 2.6, the results of tests performed by using different couple of double-talk-free signals generated as described above, are reported (in terms of average and standard deviation values). These tests were made by using the same automotive noise employed above, in condition of SNR = 20 dB and gain = 6 dB. Looking at obtained results, system performance showed to be not much influenced by the couple of selected signals for testing.

Variable SNR

More simulations have been executed in order to analyze the performance of addressed Speech Reinforcement systems as a function of the input SNR, set to the same value for both channels. As mentioned above, the noise has been

	avg	std
Channel 1		
Final AFC Mis [dB]	−1.61	0.19
Total FRLE [dB]	3.03	1.05
Final AEC Mis [dB]	−4.94	0.71
Total ERLE [dB]	11.10	2.08
ISM	0.01	0.00
LSD	0.53	0.04
Channel 2		
Final AFC Mis [dB]	−1.94	0.13
Total FRLE [dB]	4.66	0.99
Final AEC Mis [dB]	−4.04	1.02
Total ERLE [dB]	9.38	2.34
ISM	0.02	0.00
LSD	0.51	0.04

Table 2.6: Performance of the *SR w/ SUPPR. NLMS* algorithm in tests with different speech samples, Gain=6 dB, SNR=20 dB: average (avg) and standard deviation (std) values for quality indexes.

synthetically added at both front and rear microphone. Related results are reported in Table 2.7, Table 2.8 and Table 2.3 for three distinct SNR values (10 dB, 20 dB, and 40 dB), with a gain fixed at 6 dB. Again, considerable improvements can be observed when the suppressor filters are used: indeed, at such a gain value, the SR BASELINE is unable to keep stable, while the other two Speech Reinforcement systems are able to avoid the occurrence of Larsen effect. It must be therefore observed that suppressor-based systems are able to guarantee stable performance also in presence of higher gains.
It can be observed that the Speech Reinforcement performance decreases as SNR value reduces, and this is reasonable since no noise reduction action was included within the framework and, therefore, the sole noise presence negatively affects the quality of speech signals. Note that, according to the scheme depicted in Fig. 2.15, noise is amplified to the same extent of the speech signals and the SNR is not affected by the processing performed by the Speech Reinforcement system. Clearly, environmental noise will add to amplified signals, reducing the perceived SNR with respect to the SNR measured at microphone positions and considered in discussion above.
Nevertheless, suppressor-based solutions are able to guarantee the overall system stability even at SNR=10 dB. As pointed out in conclusive Section, noise reduction issue is actually object of investigations and will be addressed in future works.

Double-talk

In this Section, further results are provided for tests done in presence of double-talk. Source signals, shown in Fig. 2.27, have been synthesized partially overlapping front and rear speech signals in order to produce double-talk. When double talk is detected, by means of DTD algorithm (as described in Section 2.5), the

	SR baseline	SR w/ SUPPR. NLMS	SR w/ SUPPR. FDAF
		Channel 1	
Final AFC Mis [dB]	Unstable	0.11	0.03
Total FRLE [dB]	Unstable	0.01	−0.73
Final AEC Mis [dB]	Unstable	−3.18	−3.08
Total ERLE [dB]	Unstable	5.49	4.39
ISM	Unstable	0.02	0.02
LSD	Unstable	0.49	0.51
Isolation [dB]	Unstable	2.24	1.34
		Channel 2	
Final AFC Mis [dB]	Unstable	−0.69	−0.94
Total FRLE [dB]	Unstable	4.45	3.53
Final AEC Mis [dB]	Unstable	−2.27	−2.15
Total ERLE [dB]	Unstable	1.99	1.50
ISM	Unstable	0.02	0.04
LSD	Unstable	0.51	0.52
Isolation [dB]	Unstable	6.94	6.76

Table 2.7: Gain=6 dB, SNR=10 dB.

	SR baseline	SR w/ SUPPR. NLMS	SR w/ SUPPR. FDAF
		Channel 1	
Final AFC Mis [dB]	Unstable	−1.40	−1.40
Total FRLE [dB]	Unstable	2.68	1.81
Final AEC Mis [dB]	Unstable	−4.86	−5.06
Total ERLE [dB]	Unstable	10.51	10.39
ISM	Unstable	0.01	0.01
LSD	Unstable	0.50	0.49
Isolation [dB]	Unstable	9.04	8.78
		Channel 2	
Final AFC Mis [dB]	Unstable	−1.82	−1.97
Total FRLE [dB]	Unstable	5.10	4.76
Final AEC Mis [dB]	Unstable	−3.54	−3.87
Total ERLE [dB]	Unstable	7.79	7.65
ISM	Unstable	0.02	0.03
LSD	Unstable	0.56	0.57
Isolation [dB]	Unstable	14.79	14.40

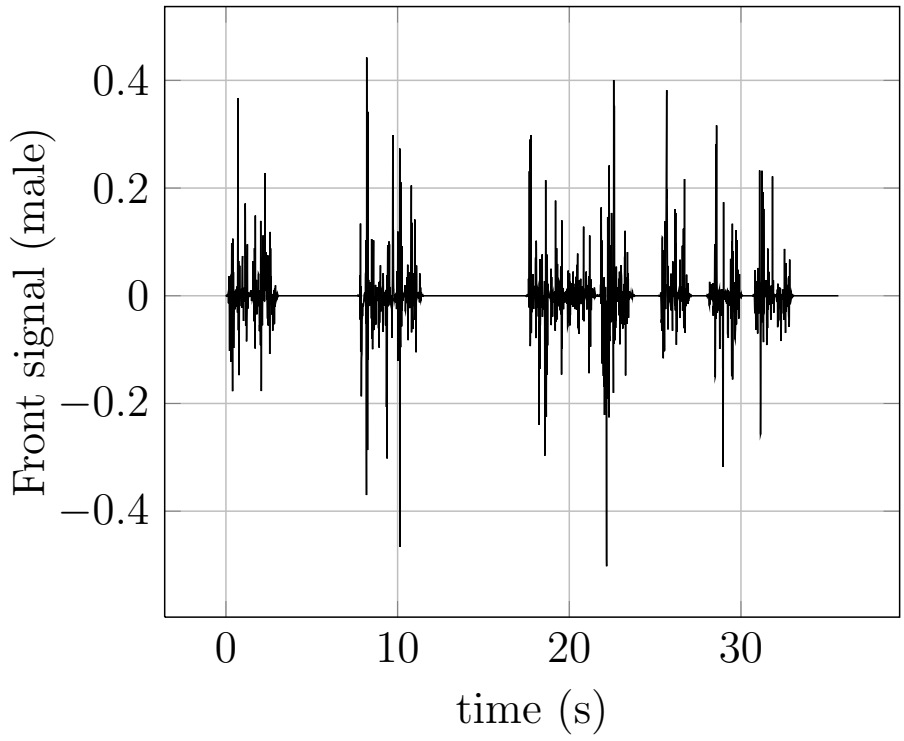
Table 2.8: Gain=6 dB, SNR=20 dB.

update of AEC filters is frozen, while the signals whitening allows to keep the AFC update enabled. For the sake of conciseness, results are reported in Table 2.9 for a single case study (Gain=3dB and SNR=20dB), as they do not present relevant differences with respect to the results relative to double-talk free scenarios, shown in previous subsections. Only a slight degradation of performance is registered due to the presence of double-talk, which hinders AEC filters update and, consequently, to the less amount of useful speech frames to be used for adaptation of filter coefficients.

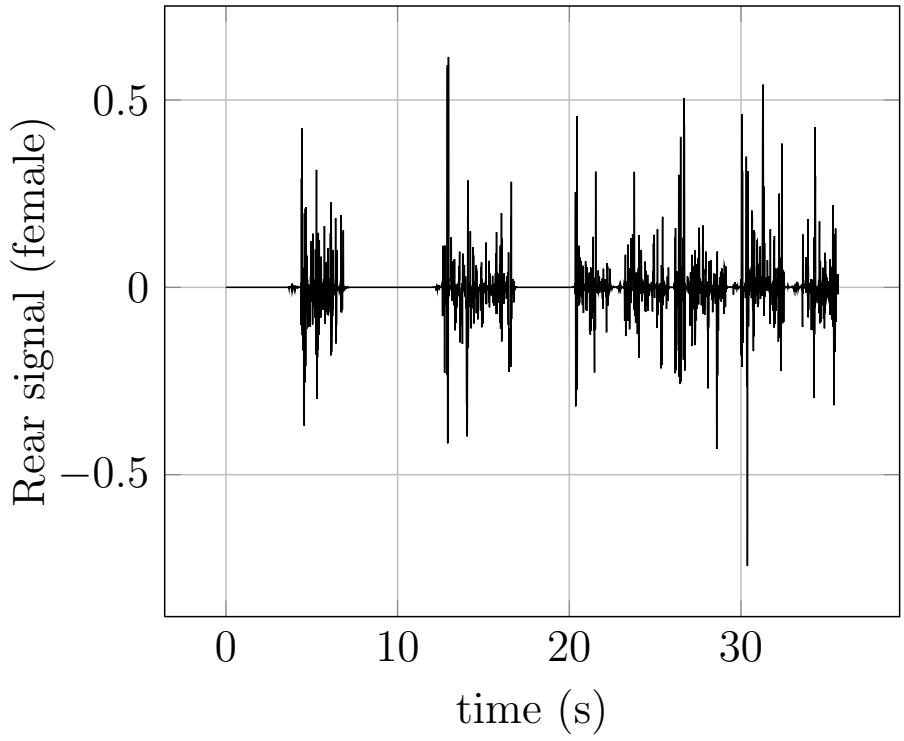
Isolation values are not reported, since, due to the double-talk, proper evaluation conditions does not occur.

	SR baseline	SR w/ SUPPR. NLMS	SR w/ SUPPR. FDAF
		Channel 1	
Final AFC Mis [dB]	−1.36	−1.39	−1.45
Total FRLE [dB]	1.14	2.48	2.01
Final AEC Mis [dB]	−3.36	−3.65	−3.89
Total ERLE [dB]	10.32	9.30	9.01
ISM	0.02	0.01	0.01
LSD	0.47	0.54	0.53
		Channel 2	
Final AFC Mis [dB]	−1.17	−1.30	−1.49
Total FRLE [dB]	3.26	3.41	3.58
Final AEC Mis [dB]	−2.52	−2.39	−2.56
Total ERLE [dB]	8.21	7.16	7.19
ISM	0.02	0.02	0.02
LSD	0.44	0.54	0.57

Table 2.9: Gain=3 dB, SNR=20 dB, Double-Talk.



(a)



(b)

Figure 2.27: Channel 1 (a) and channel 2 (b) speech signals employed to sim-

Time-Varying IRs

As appeared from simulations discussed above, SR w/ SUPPR. NLMS system is the one having the best performance. That is why it has been used here to evaluate Speech Reinforcement system behavior in presence of IR changes. The system is expected to be responsive and thus able to quickly adapt its parameter values to face the new conditions of acoustic environment. The effect of time-varying IRs on system performance has been evaluated by abruptly changing all the six IRs commented in Section 2.6.2 and used for previous simulations. Speech signals are the same depicted in Fig. 2.19. The change in IRs occurs at approximatively $t = 34s$. The IR perturbation has been synthetically simulated by adding to each IR sample a random value characterized by a normal distribution and it has been accomplished proportionally to a certain percentage of the original IR sample value itself. Three percentages have been considered for computer simulations, i.e. 20%, 60%, 100%.

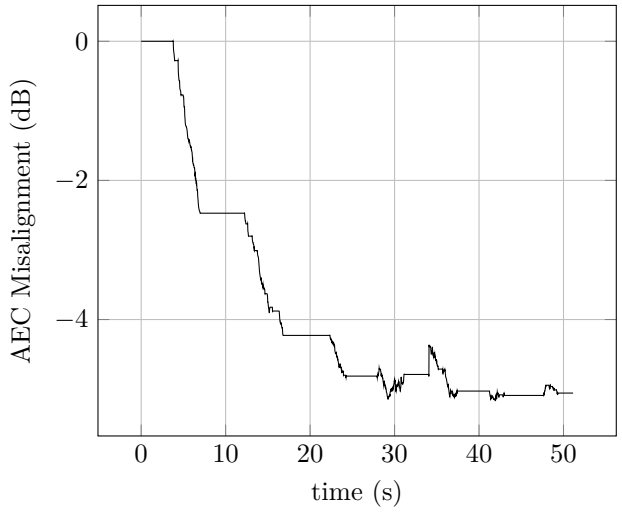


Figure 2.28: AEC misalignment with 20% perturbed IR.

Figg. 2.28-2.30 and Figg. 2.31-2.33 respectively reports the AEC and AFC Misalignment plots (all relative to channel 1), for the three IR perturbation case studied (with Gain equal to 6dB and SNR=20dB). It can be easily observed that the addressed Speech Reinforcement system is able to positively and rapidly respond to simulated environmental acoustic changes. Of course, the higher is the perturbation, the more difficult the IR tracking is. However, even when the degree of IR perturbation is relevant as in the 100% case study, the system shows a remarkable responsiveness. Behavior of other quality indexes follows the one here commented for AEC and AFC Misalignment, and it is therefore not detailed further.

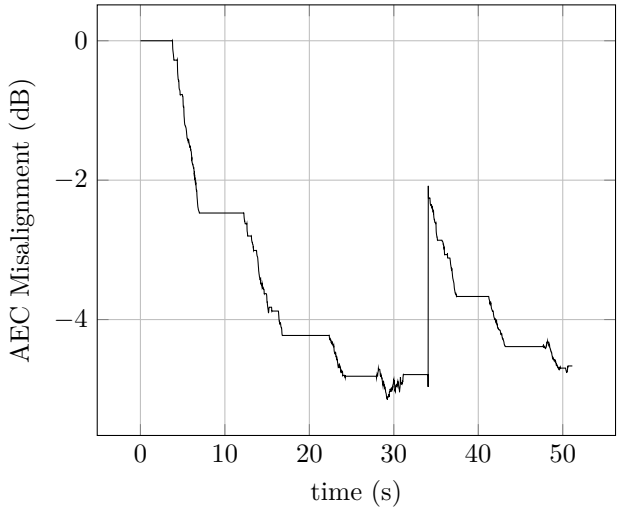


Figure 2.29: AEC misalignment with 60% perturbed IR.

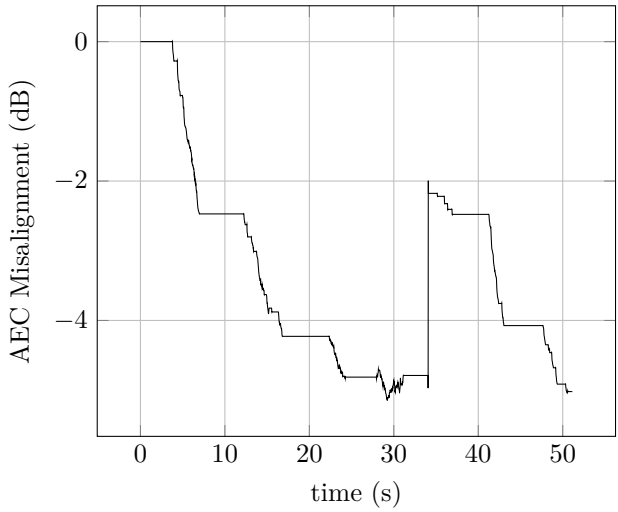


Figure 2.30: AEC misalignment with 100% perturbed IR.

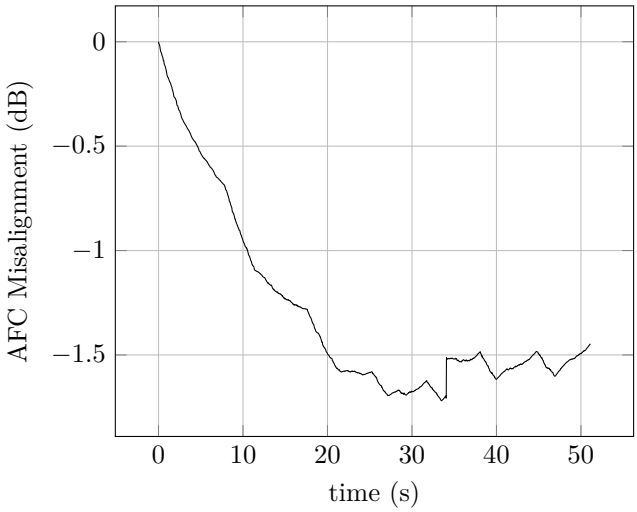


Figure 2.31: AFC misalignment with 20% perturbed IR.

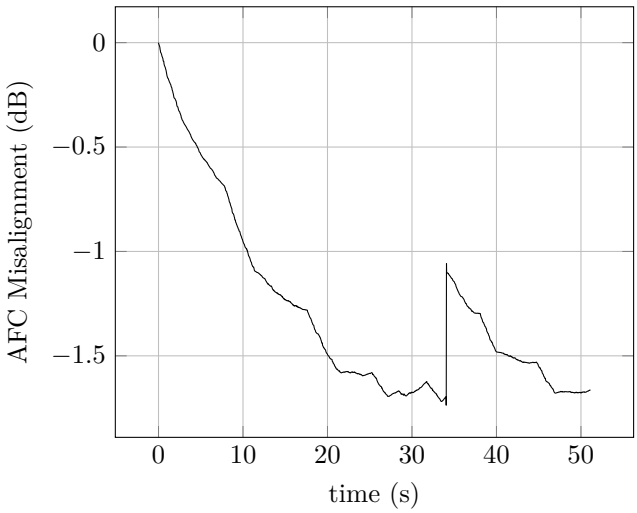


Figure 2.32: AFC misalignment with 60% perturbed IR.

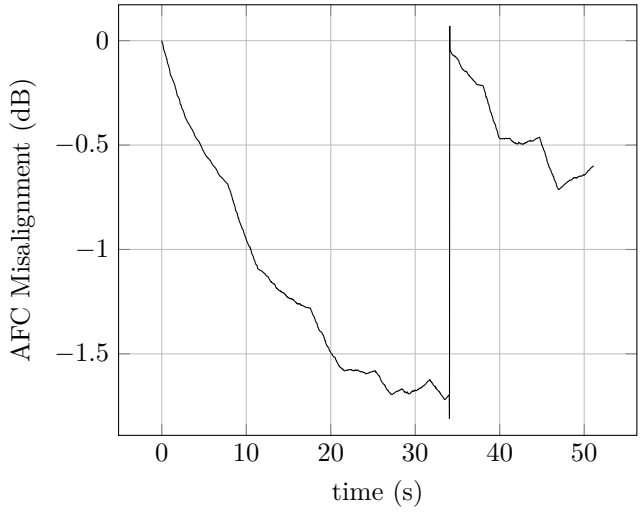


Figure 2.33: AFC misalignment with 100% perturbed IR.

Real Car Noises

Real car noises have been also used in order to carry out more realistic computer simulations. Noise signals have been recorded in the aforementioned CarLab in following acoustic conditions:

- Quiet environment (engine off, fan off, closed windows);
- Fan noise (engine off, fan on running at the max strength, closed windows);
- 50 km/h (uniform car speed, fan off, closed windows);
- 90 km/h (uniform car speed, fan off, closed windows);
- 130 km/h (uniform car speed, fan off, closed windows).

Recorded noises have been synthetically added to the same speech signals used in previous simulations and SNRs values have been set in compliance with SNRs attained in noisy speech recordings performed in the CarLab and corresponding to acoustic conditions above. These values are the following:

- SNR quiet environment = 34 dB (external background noise is audible);
- SNR fan noise = 14.5 dB;
- SNR 50 km/h = 12 dB;
- SNR 90 km/h = 11 dB;

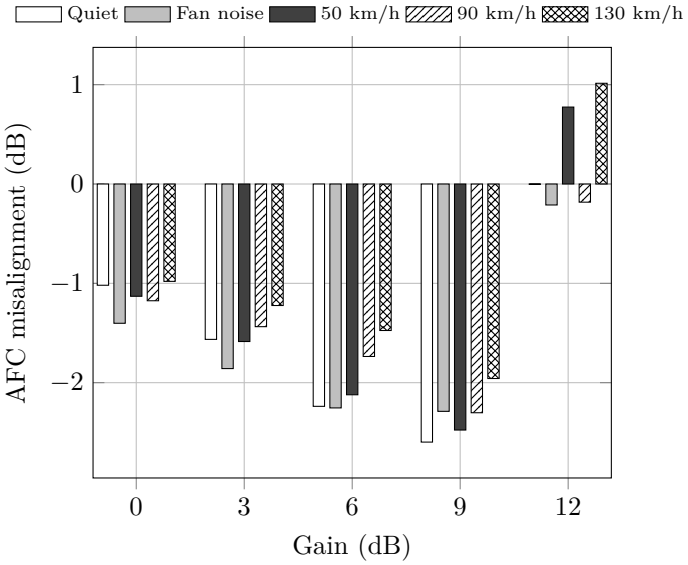


Figure 2.34: AFC misalignment in channel 1.

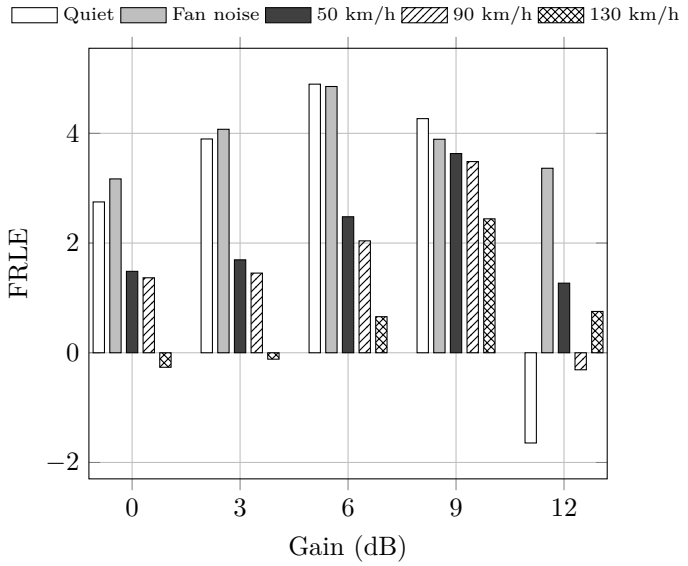


Figure 2.35: FRLE in channel 1.

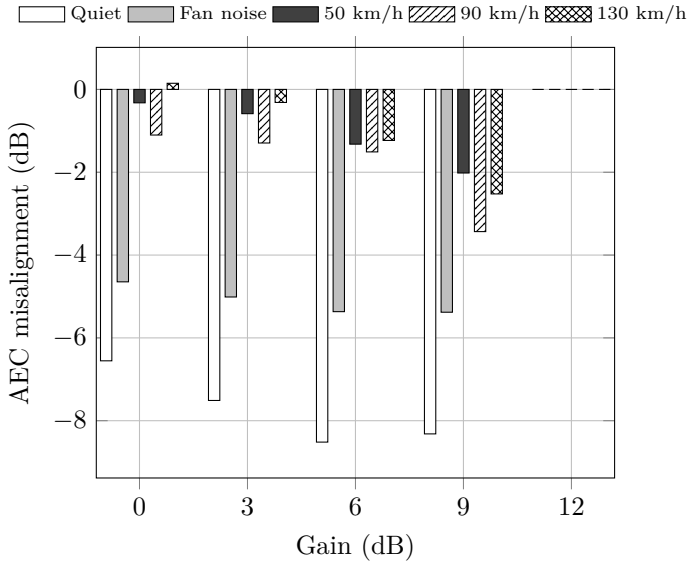


Figure 2.36: AEC misalignment in channel 1.

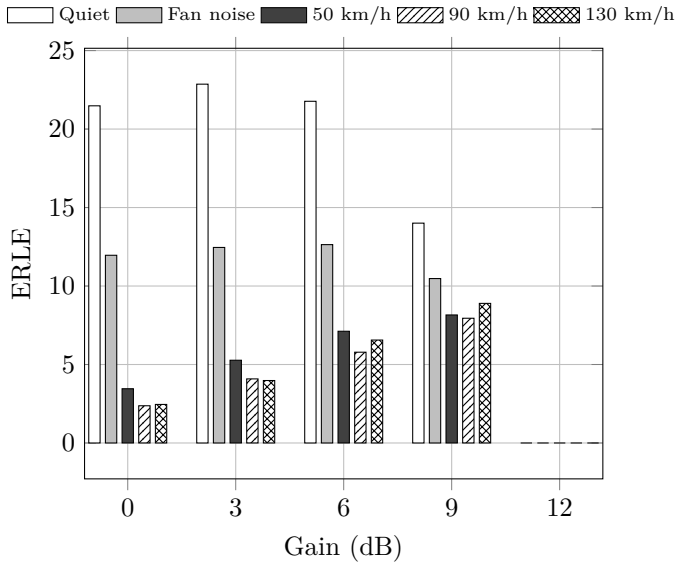


Figure 2.37: ERLE in channel 1.

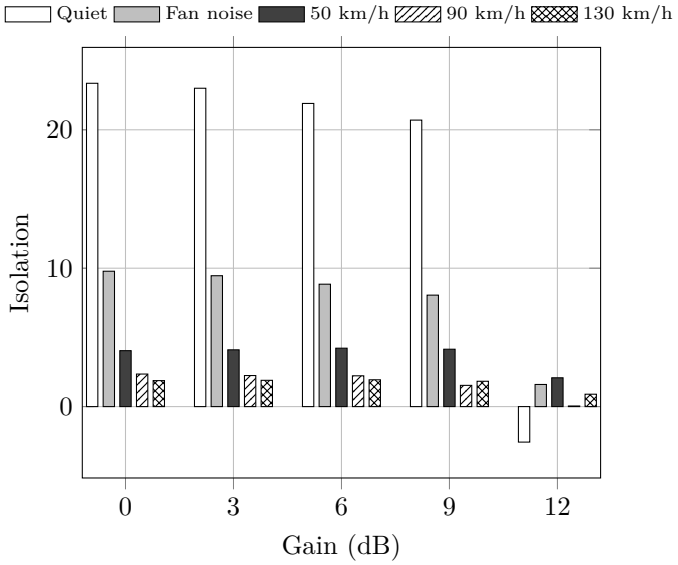


Figure 2.38: Isolation on channel 1.

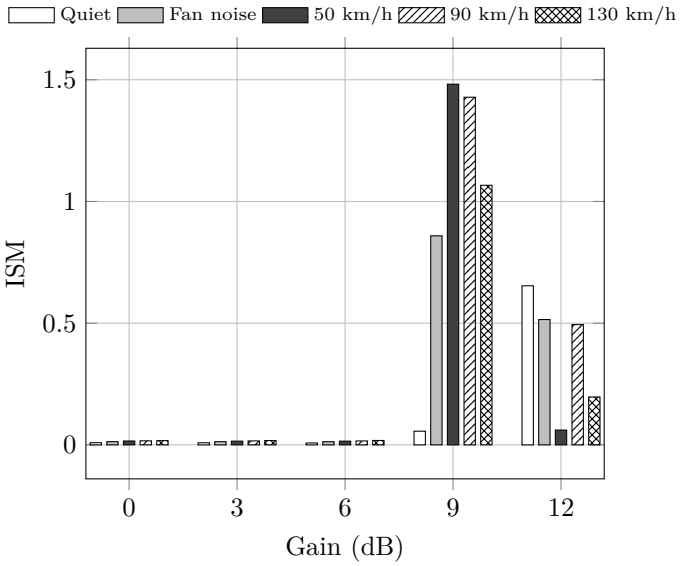


Figure 2.39: ISM on channel 1.

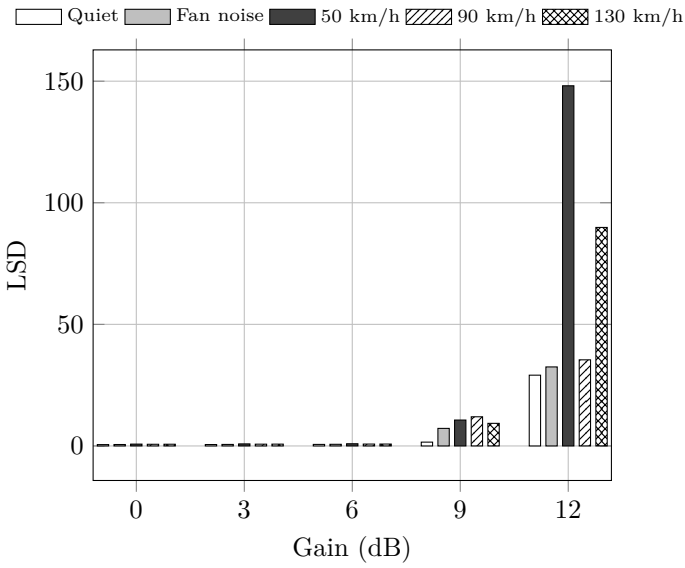


Figure 2.40: LSD on channel 1.

- SNR 130 km/h = 10.5 dB.

Also in this case, SR w/ SUPPR. NLMS approach is the one addressed for simulations. Gain values have been varied from 0 to 12dB with uniform steps of 3dB. Figg. 2.34-2.40 report the plots for all quality indexes (always relative to channel 1), for the five addressed acoustic conditions. Results show how the system performance decreases as the gain increases (specially if we look at the Isolation and the ISM and LSD values), but no stability issues arise, even in correspondence of the lowest SNR condition (car moving at 130km/h). This behavior is consistent with the one registered in previous computer simulations involving the sole noise from NOISEX database, and seems to confirm the effectiveness of the proposed Speech Reinforcement system.

Influence of Adaptive Filtering Step-sizes on System Performance

Since real-time systems might have to cope with fast varying conditions, sometimes relatively large step-sizes, to speed up filters adaptation, could be helpful. In this section it is shown how the systems behaves when the AFC and AEC step-sizes are increased with respect to the values used for the tests described in previous sections. New simulations have been performed running the algorithm *SR w/ SUPPR. NLMS* with a gain of 6 dB and same automotive noise employed above, with an SNR equal to 20 dB. Tested step-sizes values, reported in Table 2.1, are 1, 2, 4, 6, 8 and 10 times, respectively, the values used in previous simulations. These new values are denoted in Figg. 2.41-2.44 as *x1*, *x2*,

x4, *x6*, *x8*, *x10*.

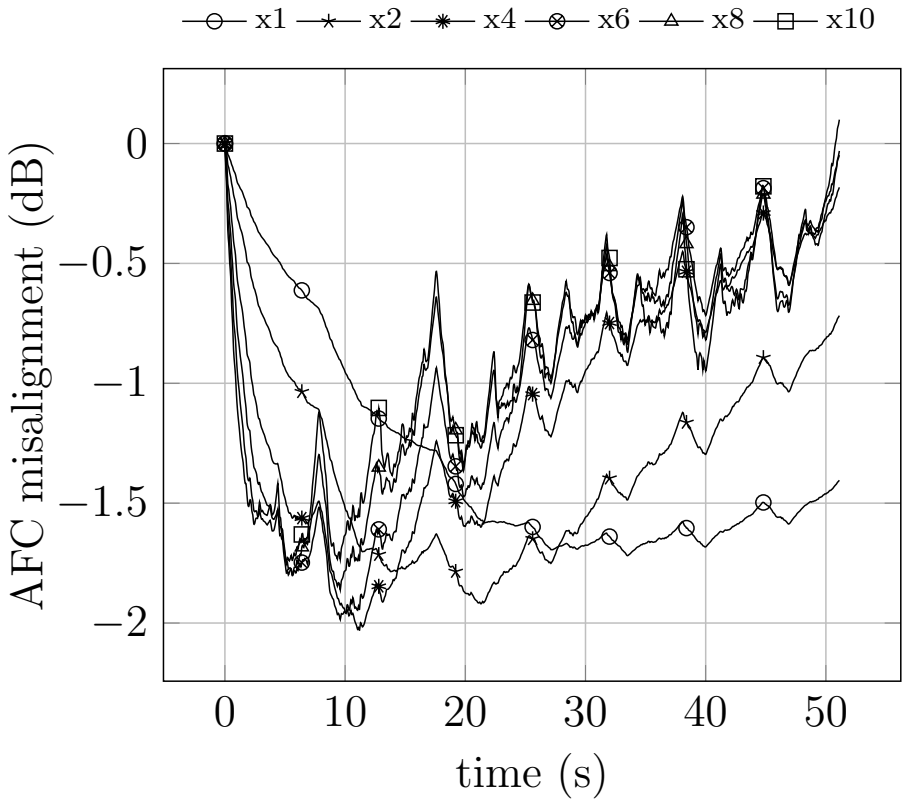


Figure 2.41: AFC misalignment in channel 1 by using different adaptive filtering stepsize values.

The results reported in Fig. 2.43 and Fig. 2.44 show that feedback adaptive filtering increases its performance (also in terms of convergence rate) as the step-size increase, whereas the echo reduction does not seem to be enhanced by the step-sizes increment (a part from the *x2* case study). In particular, it is interesting to observe that the AFC misalignment increases as the step-size increases (see Fig. 2.41), in contrast to the FRLE behaviour: this can be explained by the fact that the combination of the two joint adaptive stages within the PEM-AFROW algorithm allows to achieve good feedback cancellation performance also when the feedback path identification capability is reduced. It can be underlined that this does not occur in the echo cancellation case, where the ERLE behaviour is congruent with the misalignment one, shown in Fig. 2.42. Informal listening tests have confirmed that the echo presence is more relevant when larger step-size values are employed. All reported figures are related to one channel: similar considerations can be made for the other one.

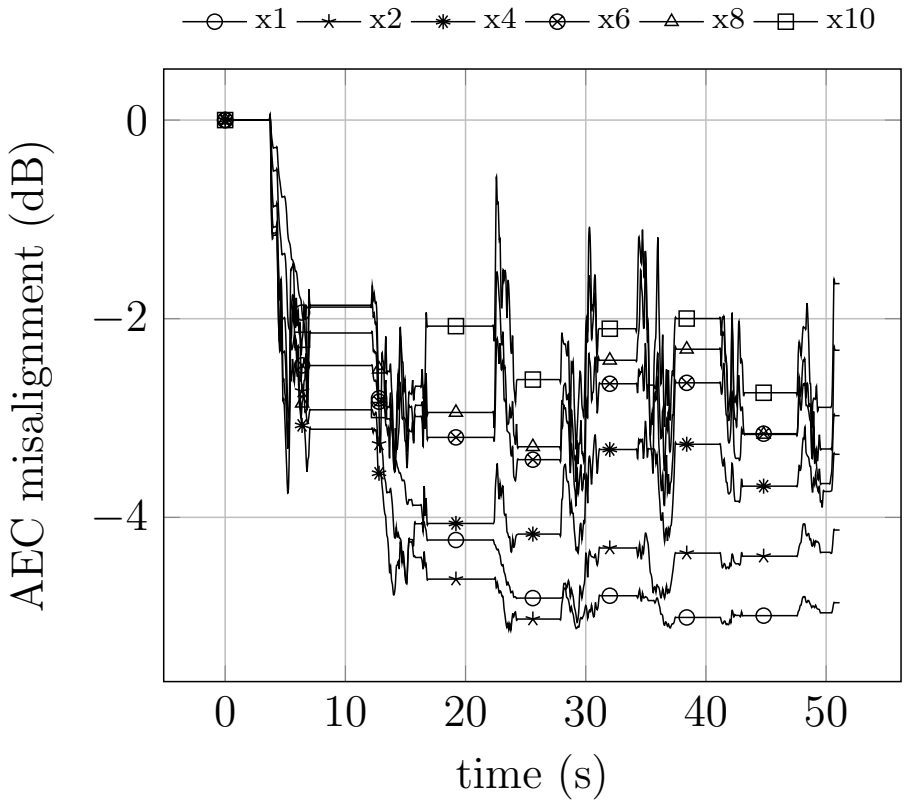


Figure 2.42: AEC misalignment in channel 1 by using different adaptive filtering stepsize values.

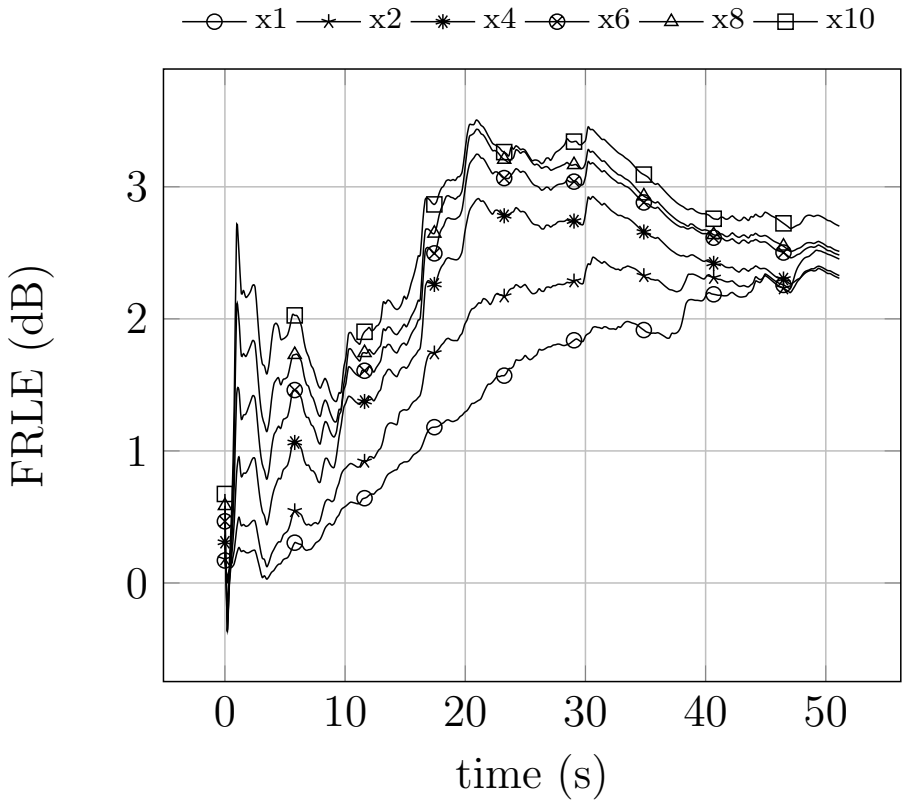


Figure 2.43: FRLE in channel 1 by using different adaptive filtering stepsize values.

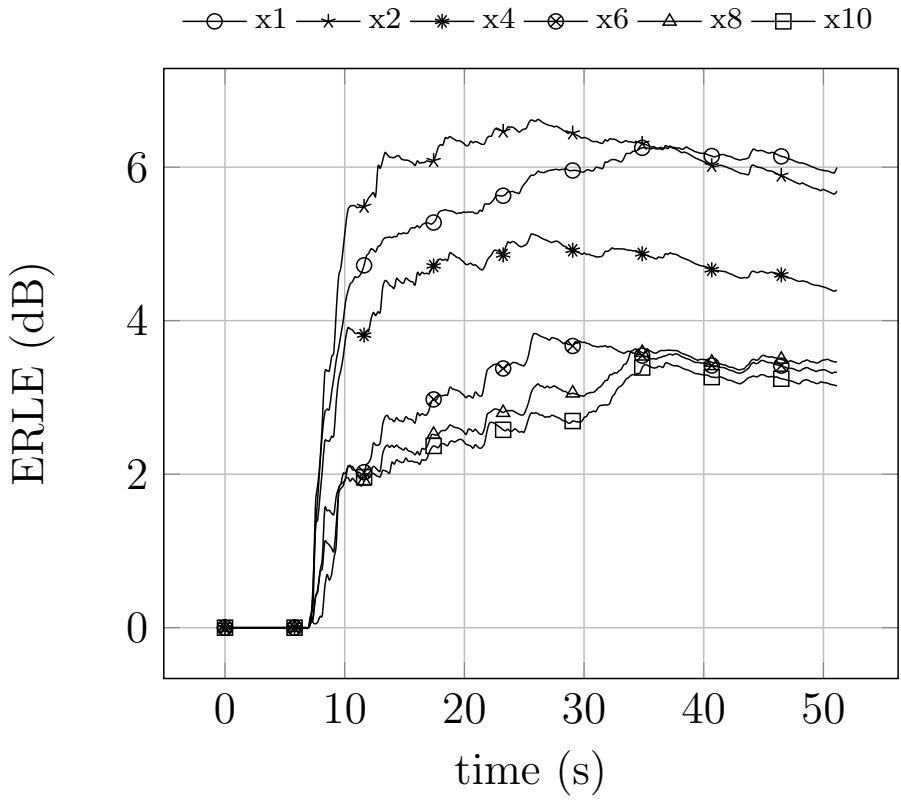


Figure 2.44: ERLE in channel 1 by using different adaptive filtering stepsize values.

Future works will be targeted to perform a fine-tuning of all step-sizes parameters looking at the system as a whole, in order to get an improved balance between the two feedback/echo reduction and adaptive filtering convergence rates. These actions will include also the investigation of suitable variable step-size strategies.

2.6.3 Computational Cost

A parametric evaluation of computational cost can be inferred from Section 2.5. Some general guidelines for the algorithmic complexity evaluation are extracted from inherent literature [36], i.e., the cost of real operations (products, sums, etc.) is considered equivalent, while one complex product costs as much as six real operations and one complex sum costs as much as two real operations. Furthermore, one real FFT is assumed to cost $2N_{DFT}\log_2 N_{DFT} - 4N_{DFT} + 6$, while one real IFFT costs $2N_{DFT}\log_2 N_{DFT}$.
The entire Dual-Channel system comprises four PB-FDAFs, two suppressors, one PEM-AFROW algorithm and two VADs. Table 2.10 shows the number of real operations per sample required by each block of the Speech Reinforcement system, together with the total cost of the SR w/ SUPPR. NLMS and SR w/ SUPPR. FDAF implementations. The column "MOPS" reports the Millions of Operations Per Second for a sampling frequency of 16 kHz, block size $L_f = 128$ and FFT length $N_{DFT} = 256$.

Algorithm	Number	Number of real operations per sample	Actual MOPS
PB-FDAF	4	$[(6N_{DFT} + 4L_f)\log_2 N_{DFT} + (4N_{DFT} + 8)L_f - 8N_{DFT} + 12]/L_f$	73.22
Suppressor NLMS	2	405	12.96
Suppressor FDAF	2	$4N_{DFT}/L_f \log_2 N_{DFT} + 6/L_f + 3$	2.15
VAD	2	$28N_{DFT}/L_f$	1.79
Whitening filters	2	18	0.58
Total w/ NLMS		$(6N_{DFT}\log_2 N_{DFT} + 20N_{DFT} + 12)/L_f + 4N_{DFT} + 4\log_2 N_{DFT} + 453$	88.54
Total w/ FDAF		$(10N_{DFT}\log_2 N_{DFT} + 20N_{DFT} + 18)/L_f + 4N_{DFT} + 4\log_2 N_{DFT} + 51$	77.74

Table 2.10: Computational cost of the Speech Reinforcement system for the different implementations of the suppressor.

The most computationally expensive blocks of framework are the PB-FDAFs, which in the current implementation require 73.22 MOPS for the four adaptive filters. More efficient implementations, that exploit FFT and FDAF filters properties may be used to significantly decrease the computational cost.
The suppressor section has different costs depending on the adaptive filter implementation. In SR w/ SUPPR. NLMS the total cost is the same of a common NLMS adaptive filter, which considering an 80 taps long filter, results in 405 operations per sample. In SR w/ SUPPR. FDAF implementation, FDAF approach helps reducing the cost to $4N_{DFT}\log_2 N_{DFT} + 6 + 3L_f$ per block, i.e., 2.15 MOPS for the two suppressors in the current implementation.
The whitening process in PEM-AFROW algorithm must be kept low, as the

Levinson-Durbin algorithm, employed to estimate the AR process coefficients, has a $O(P^2)$ complexity, leading to a total $P^2 + 2P + 3$ operation per sample. In current implementation $P = 3$, and the total cost is of a mere 18 OPS.
Finally each VAD requires $28N_{DFT}/L_f$ operations per second, for a total of 1.79 MOPS.
To resume, the computational cost of the proposed solution with FDAF suppressor filters is 77.45 MOPS given the current system parameterization. This results in a reduction of computational burden of about 12.25% with respect to SR w/ SUPPR. NLMS solution.

Chapter 3

Acoustical Comfort Enhancement Through Active Noise Control

Through the use of an adaptive/control algorithm, Active Noise Control (ANC) systems attempt to reduce a disturbance noise by generating a similar sound having same amplitude but opposite phase, resulting in an acoustical destructive interference that can theoretically bring disturbance noise level to zero. As explained in [37, 38], two different strategies can be applied to achieve this goal:

- A feedforward control strategy, in which a reference signal, coherent with disturbance noise, is sensed before noise propagates through the environment reaching the error sensor location.
- A feedback control strategy, in which the ANC system attempts to cancel disturbance noise without the advantage of a sensed reference signal, virtually reconstructing it from the knowledge of secondary path transfer function and sensed error signal.

One of the most common adaptive algorithms used in feedforward systems is the Filtered-x Least Mean Square (FxLMS), directly derived from the Least Mean Square (LMS) algorithm. Several modified/improved versions of the FxLMS were discussed in past years, such as in [39, 40], in order to cope with noises of different nature, improving robustness and performance of the original algorithm. Considering that tonal noises, generated i.e. by transformers, engines and fans, are frequently present in everyday life environments, and that because of their periodicity they feel more annoying for human auditory system than broadband noises, narrow-band approaches have been widely investigated in the past, as reported in [41, 42, 43]. Furthermore, narrow-band ANC systems are easily implementable, require low computational complexity and result in good noise cancellation. Indeed, each tone can be obtained by a combination of two phase quadrature signals (e.g. sine and cosine waves) and reference signals can be synthetically generated once the noise fundamental frequency is

detected [44]. However, when critical conditions occur, common two-taps algorithm might be led to instability [45], resulting in an emphasis of those tones that are meant to be canceled and causing a heavy comfort degradation and potential hardware saturation. When this happens, algorithm has usually no clues on what is going on and it is unable to recover from instability, leading to an irremediably compromised comfort in the environment.

In this chapter, after a brief description of classic solutions form the literature, a novel narrow-band FxLMS algorithm is proposed in Section 3.2. This new approach is based on the idea of adapting only one tap per tone and derives the other one from the knowledge of the first tap value and the amplitude of the noise tone itself, which is supposed to be time-invariant and a-priori known. The proposed approach is suitable because in most cases the noise amplitude remains the same over time for every harmonic order, and adapting only anti-noise signal phase is consequently sufficient to produce a destructive interference.
In Section 3.4, moreover, approaches based on Kalman filter theory are described as well. Although the computational burden is higher for these solutions, continuous progress in DSP hardware resources is making them more and more interesting as time goes by, also considering the good performance and low sensitivity to initial tuning that they show.

3.1 Filtered-x Least Mean Squares algorithm

FxLMS is one of the most popular adaptive algorithms derived from LMS. It distinguishes from original LMS algorithm for the addition of a compensation mechanism for undesired effects due to the secondary path (i.e. the acoustic path from the loudspeaker to the error microphone). In Section 3.1.1 classic wide-band FxLMS algorithm is described, then a two-taps narrow-band variant is introduced in Section 3.1.2.

3.1.1 Wide-Band FxLMS

In ANC systems, secondary path transfer function introduces a time misalignment between error and reference signals. As it can be seen from Fig. 3.1, changes in the control filter $W(z)$ will not be immediately observable on error signal, affecting it in a way that depends on secondary path transfer function $S(z)$ characteristics (spectral magnitude, phase, group delay, etc.). This will generally lead standard LMS algorithm to instability, making it not suitable for the purpose. FxLMS algorithm, whose operation block scheme is shown in Fig. 3.2, tries to mitigate this problem by pre-filtering reference signal $x(n)$ with an estimate $\hat{S}(z)$ of $S(z)$, in order to realign in time the sensed reference

signal with error signal, avoiding previously mentioned misalignment problems [37].

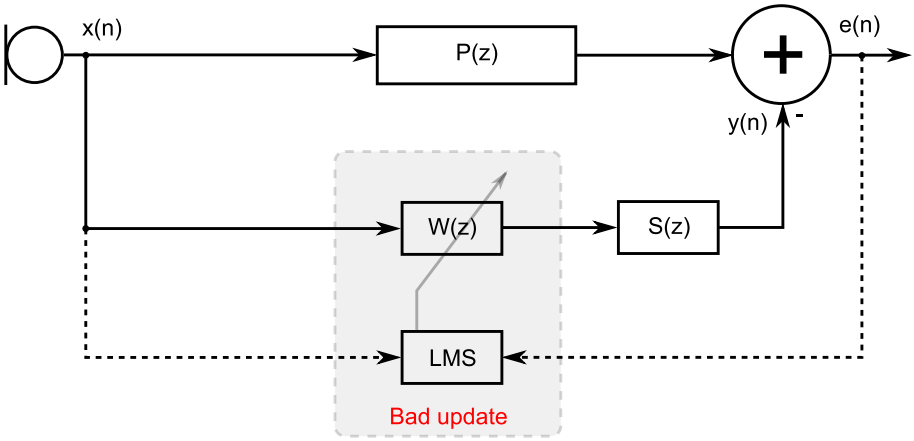


Figure 3.1: Standard LMS update scheme. Adaptive filter taps update is not consistent due to the presence of the secondary path transfer function $S(z)$.

Secondary path estimate $\hat{S}(z)$ can be measured either in advance, using a separate adaptive filter in a dedicated identification setup, or using an online estimation method as described in [46] and [47]. Adaptive filter taps $\mathbf{w}(n)$ (inverse z-transform of $W(z)$) are updated accordingly to the following equation:

$$w_i(n+1) = w_i(n) + \mu x_f(n-i)e(n), \quad i = 0, 1, ..., N \tag{3.1}$$

where $x_f(n)$ is the reference signal $x(n)$ filtered with estimated secondary path transfer function $\hat{S}(z)$, μ is the update step-size, $e(n)$ is the sensed error signal and N is the adaptive filter length (number of taps).

3.1.2 Narrow-Band FxLMS

In such situations in which disturbance noise is (mainly) of tonal nature, a narrow-band approach usually suits best than wide-band ones. Based on FxLMS algorithm seen in Section 3.1.1, Narrow-Band FxLMS (NB-FxLMS) algorithm exploits the sum of two weighted orthogonal signals (such as sine and cosine waves) in order to obtain the destructive interference of a single tone. Those orthogonal signals are also used as reference signals to feed FxLMS algorithm.

The updating formula for the two taps of a single canceling tone is the following:

$$w_k(n+1) = w_k(n) + \mu x_{k,f}(n)e(n), \quad k = 0, 1 \tag{3.2}$$

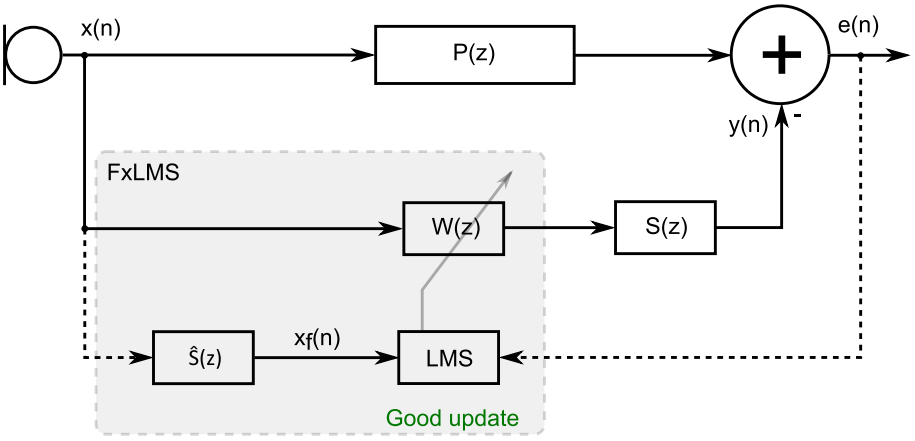


Figure 3.2: FxLMS update scheme. Filtered version $x_f(n)$ of reference signal $x(n)$ is needed to ensure adaptive filter taps update consistency.

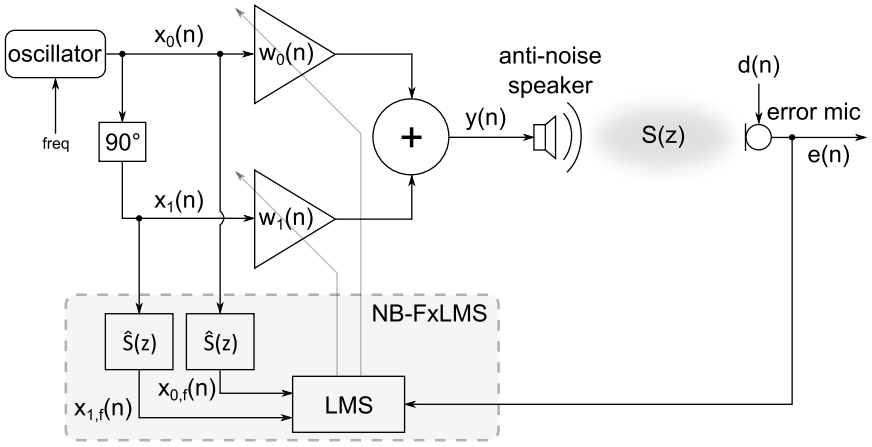


Figure 3.3: Single tone NB-FxLMS block diagram.

where k is the tap number, $x_{k,f}(n)$ is the k^{th} orthogonal signal $x_k(n)$ filtered with secondary path estimated response $\hat{S}(z)$, μ is the update step-size and $e(n)$ is the sensed error signal.
Replicating the described structure it is possible to obtain a multi-tonal canceling signal $y(n)$. Moreover, the tones that compose the noise usually have frequencies that are multiple of fundamental one. Hence, by the knowledge of the latter, it is possible to generate all the orthogonal reference signals and then cancel all the harmonics orders.

3.2 Amplitude Constrained Narrow-Band FxLMS Algorithm

Amplitude Constrained Narrow-Band FxLMS (ACNB-FxLMS) algorithm is a tap adaptation algorithm derived from NB-FxLMS described in Section 3.1.2. Observed that, in most situations, disturbance tones generated by the noise source have an amplitude that does not vary in time, rather then adapting two taps per tone in order to find the right taps combination that allows the generation of a proper anti-noise, it could be convenient to find out the disturbance tone amplitude with a preliminary measure, then use NB-FxLMS taps adaptation formula (3.2) to adapt just one of the two taps and finally, exploiting the reference signals orthogonality, infer the other tap value through the inverse formula of the Pythagorean theorem:

$$w_j(n) = \sqrt{A^2 - w_i(n)^2} \tag{3.3}$$

where A is the disturbance tone measured amplitude, w_i is the adapted tap and w_j the one to be inferred. Fig. 3.4 clarifies the relation among taps and tone amplitude.

This approach forces anti-noise tone amplitude to be never greater then A, thus implicitly avoiding instability issues when the algorithm is not converging to a proper solution. Moreover, this gives the algorithm more chances to recover from an instability situation because error signal amplitude will never diverge, reaching a maximum of $2 \cdot A$ (plus an error due to the amplitude measure itself) only in worst case scenario in which anti-noise phase exactly matches disturbance noise phase. However, always focusing the adaptation on the same tap is not sufficient, because proceeding in this way the inferred tap would never change in sign, neither when needed. Result of (3.3), indeed, is always positive. To overcame this problem, proposed algorithm exploits the fact that, since the adaptation step-size μ limits the update, only the tap having lower magnitude, approaching to zero, has the chance to change its sign in a single

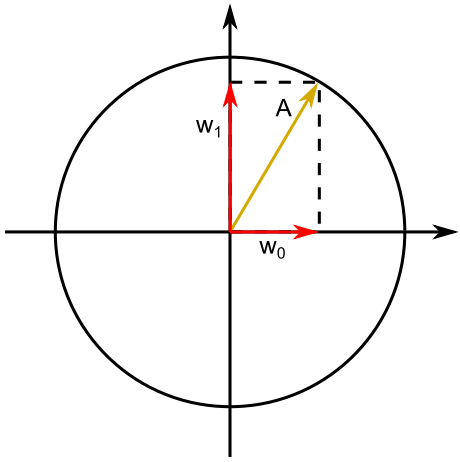


Figure 3.4: Relation among disturbance tone amplitude A and ACNB-FxLMS taps w_0 and $w1$.

update step. Therefore, the main idea is to dynamically choose which one of the two taps should be adapted (i.e. the one having the lower magnitude), inferring the other one using (3.3) to compute its modulus while maintaining the sign it had last time it was adapted, accordingly to the following equation:

$$w_j(n) = sign[w_j(n-1)]\sqrt{A - w_i(n)^2} \tag{3.4}$$

In order to make the whole system more flexible, it might be also convenient to consider an unitary amplitude during tap inference and adjust it afterwards once the two orthogonal signals have been combined together. In this way it is possible to change target amplitude by simply adjusting a final stage gain A. Nevertheless, actual amplitude of anti-noise signal must also take in account secondary path transfer function attenuation at disturbance tone frequency. Final stage gain should be consequently modified in $\frac{A}{|S(z_A)|}$ (see Fig. 3.5), where A is the measured disturbance tone amplitude at error sensor position and $|S(z_A)|$ is the secondary path amplitude at the frequency of the given tone. Thus, equation (3.3) is modified as follows:

$$w_j(n) = sign[w_j(n-1)]\sqrt{1 - w_i(n)^2} \tag{3.5}$$

Inferred tap's sign is supposed not to change again as long as its module remain the greater one. When it get lower than the other, tap choosing logic picks it and algorithm adapts its value accordingly to (3.2); this is the only situation in which its sign may change, if necessary. In order to make whole algorithm process clearer, following list summarizes what ACNB-FxLMS algorithm should

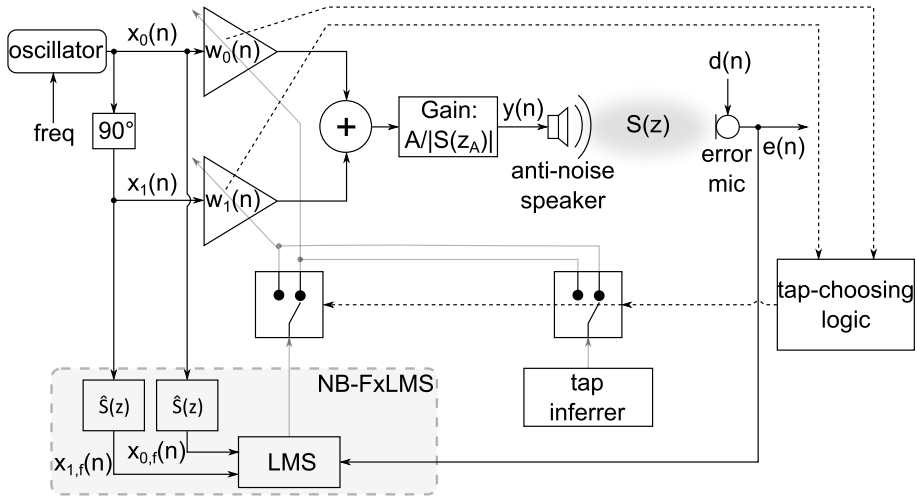


Figure 3.5: Single tone ACNB-FxLMS block diagram.

do at each iteration:

- Compare taps magnitude, to decide which one has the lower one.
- Adapt lower magnitude tap using (3.2).
- Infer other tap using (3.5).
- Weight orthogonal sine and cosine signals using taps values as weights, then sum them to compute an unitary amplitude anti-noise signal with a proper phase.
- Apply a proper gain to unitary amplitude anti-noise signal to match target amplitude A, considering also the attenuation introduced by secondary path transfer function.
- Play the anti-noise.

Fig. 3.5 shows block diagram for the cancellation of a single tone using the solution described above. As for classic NB-FxLMS approach, structure replication allows a multi-tonal cancellation.

3.3 ACNB-FxLMS Simulation Results

Several simulations have been carried out in Matlab to investigate the effectiveness of ACNB-FxLMS algorithm in multi-tone configuration. This section

describes how tests have been performed and shows how results prove the suitability of proposed algorithm.

The signal actually needed in all simulations is the noise present at error sensor position when ANC system is turned off. Considered noise source is purely tonal with a fundamental frequency of 30 Hz, consisting of four harmonic orders having the following amplitude characteristics at the sensor position:

- Order 1 (fundamental, 30 Hz), amplitude 1.0.
- Order 2 (60 Hz), amplitude 0.9.
- Order 3 (90 Hz), amplitude 0.8.
- Order 4 (120 Hz), amplitude 0.9.

As stated before, reference signals are not directly sensed, but artificially generated by the knowledge of fundamental frequency and harmonic orders. In every experiment, sampling frequency has been set to 3 kHz. Furthermore, in order to simplify the environment model, considered secondary path in the simulations consists of a pure delay of 30 samples and a gain of -3.1 dB. Considering that the initial phase delay $\Delta\phi$ of noise signal with respect to synthetic reference signals can affect the system performance, all tests have been repeated for $\Delta\phi = \frac{1}{4}\pi$, $\Delta\phi = \frac{3}{4}\pi$, $\Delta\phi = \frac{5}{4}\pi$ and $\Delta\phi = \frac{7}{4}\pi$. These values have been chosen in order to stress as much as possible the proposed ACNB-FxLM algorithm and prove its effectiveness. Indeed, with chosen initial phase delay values, both adaptive taps of each harmonic order end up having the same magnitude when the algorithm approaches convergence. Hence, the tap-choosing logic is subjected to a hard work, continuously switching the inferred tap as the difference between the two gets low. Same experiments have been repeated for both the proposed ACNB-FxLMS algorithm and the classic NB-FxLMS solution, both tuned with a step-size equal to 0.6. Tests results, presented in Fig. 3.6 and Fig. 3.7, show the error signals RMS level computed with an integration time of 100 ms. In Fig. 3.6 error signal RMS level curves obtained using classic NB-FxLMS algorithm results paired for initial phase delays of $\frac{1}{4}\pi$ and $\frac{5}{4}\pi$, and for $\frac{3}{4}\pi$ and $\frac{7}{4}\pi$. This is because a disturbance tone sign inversion does not affect convergence speed of NB-FxLMS algorithm. The tap-choosing logic non-linearity does not make this true for the ACNB-FxLMS algorithm as well but, as shown in Fig. 3.7, its adaptive capabilities are close to classic approach.

Fig. 3.8, shows an instability occurrence simulation, here achieved neglecting the secondary path effect in the update formulas. It can be observed that, while the classic NB-FxLMS error signal diverge, the ACNB-FXLM keeps it bounded, avoiding an excessive emphasis on the noise tones. This is extremely important for real implementations because it allows to avoid hardware issues and excessive comfort degradation.

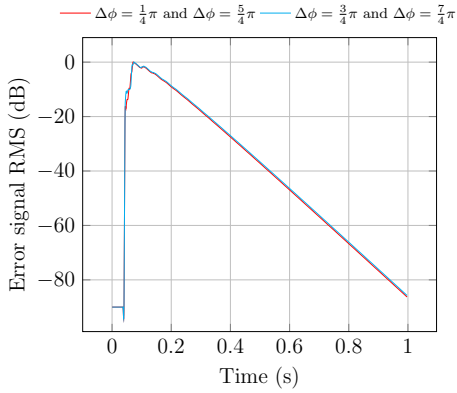


Figure 3.6: Error signal RMS level versus time for the classic NB-FxLMS algorithm in four different initial phase conditions.

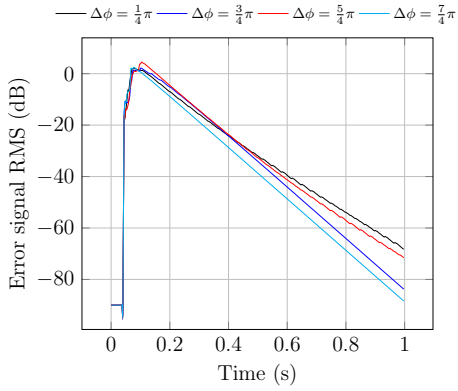


Figure 3.7: Error signal RMS level versus time for the ACNB-FxLMS algorithm in four different initial phase conditions.

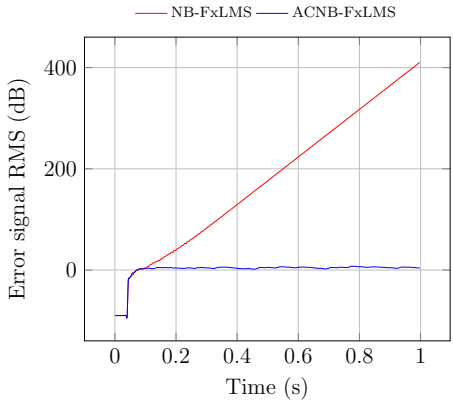


Figure 3.8: Error signals in instability condition cause by a lack of secondary path compesation.

Section concludes with an investigation on how algorithm's behavior changes when an estimate mistake affects the information on disturbance tone amplitude.

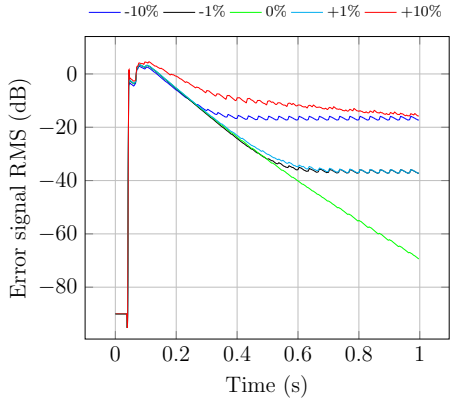


Figure 3.9: Error signals using ACNB-FxLMS in five different case of mistakes on noise amplitude information. Mistakes are introduced equally on each harmonic order.

Fig. 3.9 shows simulation results in 5 different conditions, intentionally mistaking the amplitude information of each harmonic order by -10%, -1%, 0%, $+1\%$ and $+10\%$. The introduced defect bounds the maximum achievable noise cancellation, but the algorithm still works properly succeeding to cancel part of the noise. Obviously, the more accurate information on disturbance tones

amplitudes is, the greater is noise cancellation.

3.4 Kalman Filter in ANC

Kalman filter has been widely studied in past decades because of its simplicity and robustness. However, its feasibility in ANC systems has been limited by its high computational complexity. Thanks to advances in digital computing, the interest in Kalman filter gradually increased in the years, but a multi-channel wide-band solution based on this algorithm could still be troublesome for most Digital Signal Processors (DSP) capabilities. However, in many real-world scenarios, the acoustical ambient noise is mainly of tonal nature, and a narrow-band ANC approach can be adopted. This leads to a significant reduction in the number of necessary taps to achieve the noise cancellation goal, making the Kalman filter more suitable for practical implementations.
In this section it is explained how to fit an ANC problem into the Kalman filter theory described in Section 1.2, extending the Kalman IRs identification approach described in Section 1.2.3. Section 3.4.1 introduces a single-channel wide-band approach, while Section 3.4.2 adapts it to narrow-band applications. Section 3.4.3 extends the former to a multi-channel scenario while Section 3.4.4 derives a multi-channel narrow-band solution.

3.4.1 Single-Channel Wide-Band Solution

Kalman-based Single-Channel Wide-Band algorithm derives from similar considerations to the ones presented in Section 1.2.3. However, an ANC scenario, beside not having control input signals u_k because overall environment behavior is not governable, introduces two additional phenomena that were not present in a simple IR identification task and that complicates the solution of the identification problem. These additional phenomena are the following:

1. Inability to directly observe output signal y_k in order to obtain measure z_k, because in the attempt to cancel noise, it will be modified; consequently z_k will depend from ANC system itself.

2. Presence of an additional acoustical path $S(z)$ between the anti-noise source and the error sensor position, which delays and modifies how y_k is perceived by the transducer which measures it.

It has to be noted that, as shown in generic ANC scheme of Fig. 3.10, the noise coming from the source reaches the error sensor position through the primary path $P(z)$; however, since in an ANC system the reference signal r_k goes through both the adaptive filter $\hat{x}_k$ and the secondary path $S(z)$, it is possible to think of $P(z)$ as the cascade of two filters as well: an ideal noise

shaping filter x_k given by (3.6), whose taps are the system state variables that Kalman algorithm has to estimate, and the secondary path filter $S(z)$. This latter can be estimated separately from the former, either in advance or through an online estimation method [48, 46, 47] so that x_k is the only unknown filter left.

$$x_k = \begin{pmatrix} x_{k,1} \\ x_{k,2} \\ \vdots \\ x_{k,n} \end{pmatrix} \tag{3.6}$$

Therefore, Kalman filter goal in a single-channel wide-band ANC application

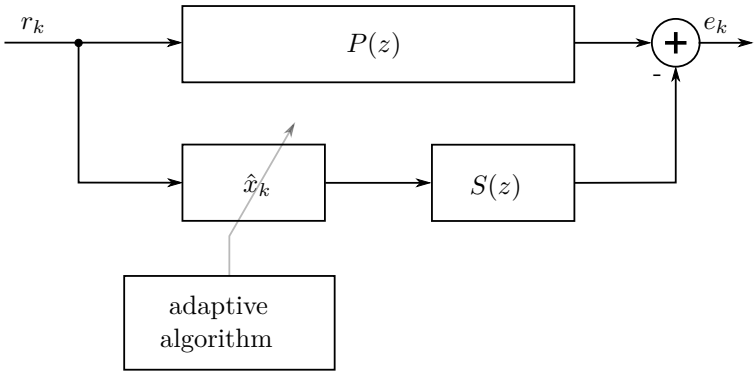


Figure 3.10: Generic ANC scheme.

is to find the estimated taps values $\hat{x}_k$ such that, thinking of $P(z)$ as the cascade of x_k FIR filter and secondary path transfer function $S(z)$, the results of the filtering of reference signal r_k by both the cascade of $\hat{x}_k$ filter and $S(z)$ transfer function, and by the primary path transfer function $P(z)$, are as close as possible, thus minimizing the error signal e_k at error microphone position. As for the IRs identification problem described in Section 1.2.3, reference signal samples are actually used to build the matrix H_k, given by (3.7).

$$H_k = \begin{pmatrix} r_k & r_{k-1} & \cdots & r_{k-n+1} \end{pmatrix} \tag{3.7}$$

However, the output y_k of ideal filter x_k is not accessible because the primary path $P(z)$ is actually uncut; moreover, as noticed before, ANC system modifies y_k in its attempt of reducing the disturbance noise. In the meanwhile, the secondary path $S(z)$ also affects the sensed signal e_k at the error sensor position, which is given by (3.8).

$$e_k = (y_k + \hat{y}_k) \otimes \mathbf{s}_k + v_k = (y_k \otimes \mathbf{s}_k) + (\hat{y}_k \otimes \mathbf{s}_k) + v_k \tag{3.8}$$

$\mathbf{s}_k$ is the secondary path impulse response, $\hat{y}_k$ is the adaptive filter output, v_k is the measurement inaccuracy due to the error sensor non-idealities and $\otimes$ indicates the convolution operation. For the above reasons a direct observation/measurement of y_k is not feasible, and the only chance to determine it is by exploiting an indirect method, trying to virtually add back what the ANC system took away from it, combining in some way the other known signals and transfer functions. One possible solution is to subtract the filtered anti-noise signal $(\hat{y}_k \otimes \mathbf{s}_k)$ from e_k, exploiting a secondary path estimate $\hat{S}(z)$ which is supposed to be available, to obtain a measure z_k of the filtered output signal $(y_k \otimes \mathbf{s}_k)$ given by (3.9).

$$z_k = e_k - (\hat{y}_k \otimes \hat{\mathbf{s}}_k) = (y_k \otimes \mathbf{s}_k) + (\hat{y}_k \otimes \mathbf{s}_k) - (\hat{y}_k \otimes \hat{\mathbf{s}}_k) + v_k \tag{3.9}$$

If the secondary path impulse response estimate $\hat{\mathbf{s}}_k$ is perfect (i.e. $\hat{\mathbf{s}}_k = \mathbf{s}_k$), the measure z_k will be equal to the filtered output signal $(y_k \otimes \mathbf{s}_k)$, unless the measurement inaccuracy v_k. At this point a measurement z_k has been obtained, but the issue regarding the time misalignment between this signal and the reference signal in H_k matrix due to the secondary path effect is still pending. The solution to this problem is directly inspired by the FxLMS approach described in Section 3.1 and consists in filtering H_k with secondary path estimate $\hat{S}(z)$, in order to re-align them. Fig. 3.11 shows a block diagram that summarizes the concepts just described.

Finally, all the elements that are necessary to fit the single-channel wide-band

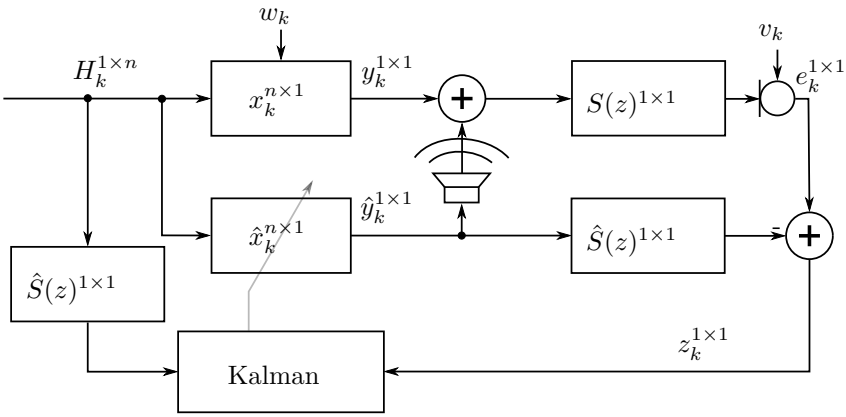


Figure 3.11: Kalman filter configuration for wide-band single-channel systems.

ANC problem into Kalman filter theory has been defined and only a good tuning of parameters Q and R is missing to make the system properly work.
It should also be noted that, for single channel solutions, y_k and z_k arrays

have the same size but, as will be explained in Section 3.4.3, this is not generally true for multi-channel configurations, where the number of speakers and microphones are not necessary the same.

3.4.2 Single-Channel Narrow-Band Solution

Kalman filter is a powerful but computationally expensive algorithm, which may require too much resources to be implemented on DSPs having limited capabilities. However, it is common to deal with situations in which the disturbance noise is (mainly) of tonal nature. In such situations, a narrow-band approach usually suits better than a wide-band one and allows to considerably save computational resources. Also considering last years advances in digital computing, this approach seems becoming more attractive as time goes by.
Single-channel narrow-band Kalman filter directly arise from the wide-band approach described in Section 3.4.1 whit the difference that this exploits the sum of two weighted orthogonal signals (e.g. sine and cosine waves) to achieve a destructive interference of a single tone. These same signals are also used as reference signals for the adaptive algorithm, which means their actual samples values will be used for the construction of the input reference matrix H_k. Fig. 3.12 shows the block diagram for a single-tone, narrow-band, single-channel ANC system using Kalman filter. For a multi-tone cancellation it will be sufficient to add, for each tone, two weights to the system state vector x_k, and the actual samples values of the two orthogonal reference signals (of the same

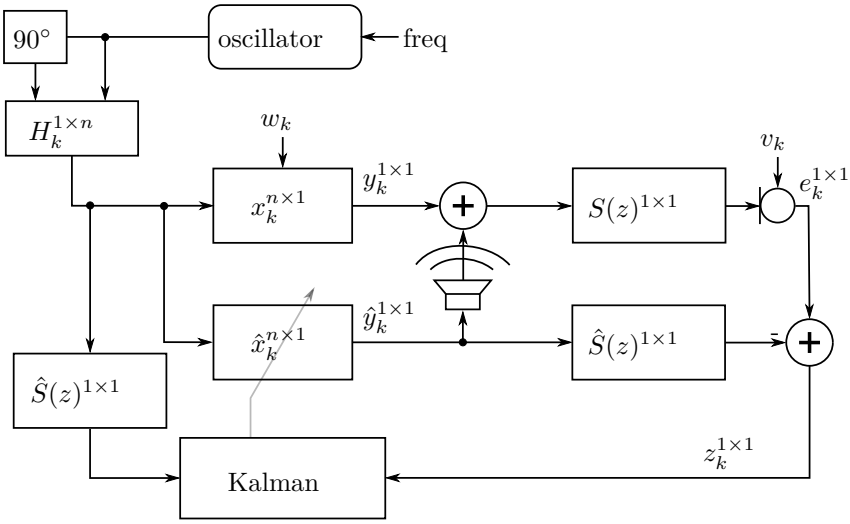


Figure 3.12: Kalman filter configuration for a single-channel narrow-band system.

frequency as the disturbance tone) into matrix H_k. Following this design, the total number of taps, i.e. the length n of the state vector x_k, is obviously equal to twice the number n_h of tones to be canceled, as shown in (3.10).

$$n = 2 \cdot n_h \tag{3.10}$$

The state of the system that Kalman filter aims to estimate, given by (3.11), is composed by the series of all the needed weights, two by two related to a certain tone.

$$x_k = \begin{pmatrix} x_{sin_1,k} \\ x_{cos_1,k} \\ \vdots \\ x_{sin_{n_h},k} \\ x_{cos_{n_h},k} \end{pmatrix} \tag{3.11}$$

The matrix H_k, instead, is given by (3.12), where $r_{sin_i,k}$ and $r_{cos_i,k}$ are respectively current samples of sine and cosine reference signals related to the i^{th} disturbance tone.

$$H_k = \begin{pmatrix} r_{sin_1,k} & r_{cos_1,k} & \cdots & r_{sin_{n_h},k}, r_{cos_{n_h},k} \end{pmatrix} \tag{3.12}$$

Moreover, tones composing the disturbance noise usually have frequencies that are multiple of a fundamental one. Hence, by the knowledge of the latter, e.g. by means of a single frequency tracker, it is possible to digitally generate all the orthogonal reference signals that are necessary to cancel the desired harmonic orders.

3.4.3 Multi-Channel Wide-Band Solution

In the attempt of distributing ANC benefits over the whole considered environment area as uniformly as possible, thus avoiding focused cancellation around a single error position, practical applications often need to deal with multiple error microphones. In order to assure this approach feasibility and/or increase ANC system performance, multiple anti-noise speakers are often used as well, resulting in a multi-channel system having a number of channels equal to $d \times m$, with d the number of speakers and m the number of microphones. Because of its matrix formulation, Kalman filter is already designed for the identification of systems having an arbitrary number of outputs y_k exploiting an arbitrary number of measurements z_k, even if in classic formulation both have the same dimension. Nevertheless, in a multi-channel system, the number of speakers d is not necessarily equal to the number of microphones m, so what is left to do is fitting the Kalman filter theory formulation to a multi-

speaker/multi-microphone ANC scenario. System outputs vector y_k length and measurements vector z_k length will be equal to the total number of speakers d and the total number of microphones m, respectively. As already mentioned in Section 3.4.1, the elements of the $d \times 1$ outputs vector y_k are not available for a direct measurement, and their observation takes place at the error sensors positions where, after the effect of the $m \times d$ secondary paths $S^{m\times d}(z)$ given by (3.13), the $m \times 1$ measure z_k is available.

$$S^{m\times d}(z) = \begin{pmatrix} S_{1,1}(z) & S_{2,1}(z) & \cdots & S_{d,1}(z) \\ S_{1,2}(z) & S_{2,2}(z) & \cdots & S_{d,2}(z) \\ \vdots & \vdots & \vdots & \vdots \\ S_{1,m}(z) & S_{2,m}(z) & \cdots & S_{d,m}(z) \end{pmatrix} \tag{3.13}$$

$S_{j,i}(z)$ represents the secondary path between the j^{th} speaker and the i^{th} microphone, placed in position (i, j) of matrix $S^{m\times d}(z)$ (i^{th} row and j^{th} column). Referring to $Y_k(z)$ as the array of the Z-transforms $Z[\cdot]$ of each output signal sequence in y_k, taken at discrete-time interval k (3.14), equation (3.15) mathematically expresses the relation between this one and z_k when the ANC system is off. Introduced superscripts emphasizes dimensions.

$$Y_k^{d\times 1}(z) = \begin{pmatrix} Z\left[y_{k,1}\right] \\ Z\left[y_{k,2}\right] \\ \vdots \\ Z\left[y_{k,d}\right] \end{pmatrix} = \begin{pmatrix} Y_{k,1}(z) \\ Y_{k,2}(z) \\ \vdots \\ Y_{k,d}(z) \end{pmatrix} \tag{3.14}$$

$$z_k^{m\times 1} = Z^{-1}\left[S^{m\times d} Y_k^{d\times 1}(z)\right] \tag{3.15}$$

$Y_{k,j}(z)$ is the j^{th} element of $Y_k(z)$ and $Z^{-1}[\cdot]$ represents the inverse Z-transform element-by-element operator. Hence, the i^{th} element $z_{k,i}$ of the measurement vector z_k is given by (3.16).

$$z_{k,i} = Z^{-1}\left[\sum_{j=1}^{d} S_{j,i}(z) Y_{k,j}(z)\right] \tag{3.16}$$

When ANC system is on, as said before, measurement z_k must be compensated by the anti-noise signals effect, and equations (3.15) and (3.16) becomes (3.17) and (3.18) respectively.

$$z_k^{m\times 1} = e_k - Z^{-1}\left[\hat{S}^{m\times d}(z)\hat{Y}_k^{d\times 1}(z)\right] \tag{3.17}$$

$$z_{k,i} = e_{k,i} - Z^{-1}\left[\sum_{j=1}^{d} \hat{S}_{j,i}(z)\hat{Y}_{k,j}(z)\right] \tag{3.18}$$

where e_k and $e_{k,i}$ are the error signals vector and its element respectively, $\hat{S}^{m\times d}(z)$ and $\hat{S}_{j,i}(z)$ the estimated secondary path matrix and its element respectively, and $\hat{Y}_k^{d\times 1}(z)$ and $\hat{Y}_{k,j}(z)$ the anti-noise signals vector and its element respectively.
Due to the effect of secondary path matrix $S^{m\times d}(z)$, measurement vector z_k is not only time misaligned with respect to reference signals in matrix H_k, but also its size is not suitable. As was for single-channel algorithm, even in the multi-channel case it is sufficient to apply secondary path filters $S^{m\times d}(z)$ to the reference signals matrix H_k to overcome both issues. Fig. 3.13 schematically represents the whole idea, reporting dimensions as superscripts.

The $n \times 1$ system state vector x_k can be interpreted as the concatenation of

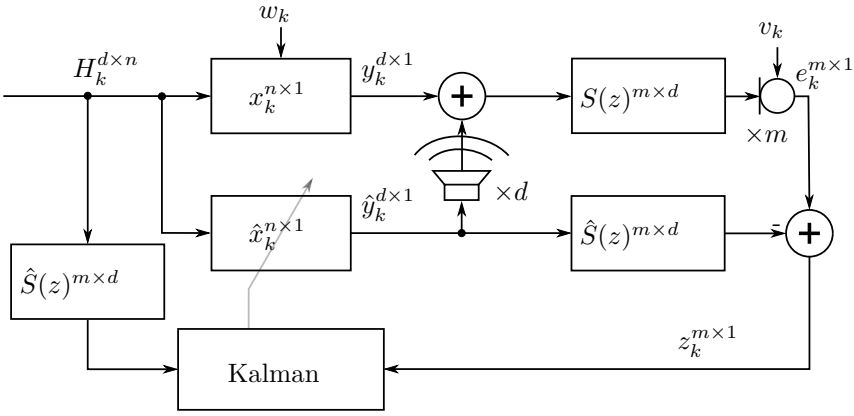


Figure 3.13: Kalman filter configuration for wide-band multi-channel systems.

d series of n_t taps, one for each output signal, with n given by (3.19).

$$n = d \cdot n_t \tag{3.19}$$

Equation (3.20) clarifies how x_k is composed.

$$x_k^{n\times 1} = \begin{pmatrix} x_{1,k}^{n_t\times 1} \\ x_{2,k}^{n_t\times 1} \\ \vdots \\ x_{d,k}^{n_t\times 1} \end{pmatrix} \tag{3.20}$$

As said before, each element of the system outputs vector y_k is related to a given series of n_t taps in x_k, and accordingly to this the $d \times n$ input reference matrix H_k has to be built assigning the i^{th} row to the i^{th} element of y_k. Consequently, i^{th} row should be zero-filled except for the elements in columns from position $(i-1) \cdot n_t + 1$ to $i \cdot n_t$, containing a reference signal delay-line of n_t samples. Equation (3.21) clarifies what has just been expressed by words.

$$H_k^{d \times n} = \begin{pmatrix} r_k^{1 \times n_t} & 0^{1 \times n_t} & \cdots & 0^{1 \times n_t} \\ 0^{1 \times n_t} & r_k^{1 \times n_t} & \cdots & 0^{1 \times n_t} \\ \vdots & \vdots & \vdots & \vdots \\ 0^{1 \times n_t} & 0^{1 \times n_t} & \cdots & r_k^{1 \times n_t} \end{pmatrix} \tag{3.21}$$

3.4.4 Multi-Channel Narrow-Band Solution

As mentioned before, using Kalman filter in ANC applications is attractive due to its good performances, but real-time execution of the algorithm requires a considerable amount of computational power and memory. Furthermore, when dealing with a multi-channel scenario, required resources are even more, leading to a burdensome algorithm. Nevertheless, a narrow-band adaptation of the multi-channel algorithm presented in Section 3.4.3 helps again in "lightening up" the whole algorithm, as was for the single-channel one. In this section, how to combine the multi-channel and narrow-band approaches, obtaining the best of both, is explained.
Secondary paths effects remains the same as for the multi-channel wide-band solution, with $S^{m \times d}(z)$ given by (3.13). The difference here is that, as explained in Section 3.4.2, a narrow-band approach exploits the sum of two weighted orthogonal signals to achieve a destructive interference of a single disturbance tone. The block diagram of the Kalman filter multi-channel narrow-band system is show in Fig. 3.14. For a single-tone cancellation, two sine-cosine weights for each of the d output signals (i.e. the number of anti-noise speakers) are needed, resulting in a total length n of the internal state vector equal to $2d$. A multi-tone cancellation considering n_h harmonic orders, hence, requires a series of $2n_h$ weights, two by two assigned to a certain harmonic order, for each of the d output signals; this leads to the general case equation of the internal state vector length given by (3.22).

$$n = d \cdot (2 \cdot n_h) \tag{3.22}$$

Accordingly to above sentences, the complete system internal state vector x_k, given by (3.24), is obtained from the concatenation of d arrays given by (3.23) containing $2n_h$ weights each and related to the d^{th} system output. Superscripts

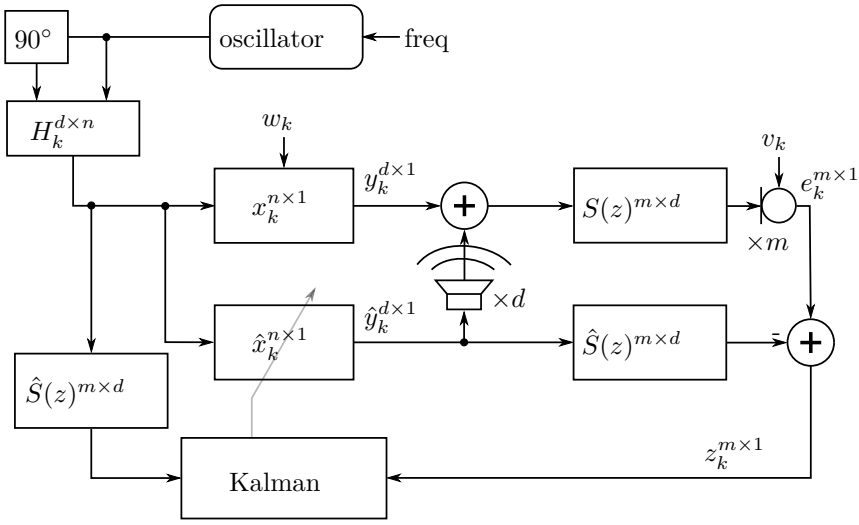


Figure 3.14: Kalman filter configuration for a multi-channel narrow-band system.

emphasize arrays' dimensions.

$$x_{i,k}^{2n_h\times 1} = \begin{pmatrix} x_{sin_1,i,k} \\ x_{cos_1,i,k} \\ \vdots \\ x_{sin_{n_h},i,k} \\ x_{cos_{n_h},i,k} \end{pmatrix} \quad i = 1, \cdots, d \tag{3.23}$$

$$x_k^{n\times 1} = \begin{pmatrix} x_{1,k}^{2n_h\times 1} \\ x_{2,k}^{2n_h\times 1} \\ \vdots \\ x_{d,k}^{2n_h\times 1} \end{pmatrix} \tag{3.24}$$

The i^{th} row of input reference matrix H_k, related to the i^{th} system output, is filled from position $(i-1)\cdot 2n_h+1$ to $i\cdot 2n_h$ with the row array $r_{h,k}$ of the $2n_h$ tonal references samples given by (3.25), and zeros elsewhere.

$$r_{h,k} = \begin{pmatrix} r_{sin_1,k} & r_{cos_1,k} & \cdots & r_{sin_{n_h},k}, r_{cos_{n_h},k} \end{pmatrix} \tag{3.25}$$

The whole matrix H_k, instead, is build as shown in equation (3.26).

$$H_k^{d\times n} = \begin{pmatrix} r_{h,k}^{1\times 2n_h} & 0^{1\times 2n_h} & \cdots & 0^{1\times 2n_h} \\ 0^{1\times 2n_h} & r_{h,k}^{1\times 2n_h} & \cdots & 0^{1\times 2n_h} \\ \vdots & \vdots & \vdots & \vdots \\ 0^{1\times 2n_h} & 0^{1\times 2n_h} & \cdots & r_{h,k}^{1\times 2n_h} \end{pmatrix} \tag{3.26}$$

3.5 Kalman Algorithm Simulation Results

A simulation environment has been developed in *Matlab* to test all the Kalman-based ANC algorithms described in Section 3.4. Moreover, to show their effectiveness, performances have been compared to those of the relative FxLMS-based counterparts.
Developed simulator has been designed to perform the following tasks:

- Generate the signal emitted by the disturbance noise source, meant to be canceled by the ANC system at error microphones positions.
- Generate all the primary and secondary path transfer functions, representing the sound propagation paths from the disturbance noise source to each error microphone and from each anti-noise speaker to each error microphone, respectively.
- Simulate the sound propagation of disturbance noise through each primary path, and of each anti-noise signal through each secondary path, by the use of a FIR filtering operation.
- Simulate the acoustical summation of all anti-noise signals with the disturbance noise occurring at each error microphone position.
- Run the identification algorithm to compute the sample values of all anti-noise signals that are meant to be played through available speakers at next time step.

For a proper operation of all Kalman-based algorithms, the simulator also performs a reconstruction of the original disturbance noise signal present at each error microphone position, exploiting (3.9) introduced in Section 3.4.1. For simplicity, a perfect knowledge of all secondary path impulse responses has been assumed in every test.
Furthermore, disturbance noise signal generator has been designed to be able to generate both wide-band and narrow-band noise types with prescribed characteristics. A zero-mean pink noise having unitary variance has been considered as disturbance noise in all the wide-band algorithms tests, while a tonal noise having a fundamental frequency of 30 Hz and composed of 1^{st}, 2^{nd}, 3^{rd}, 4^{th}

harmonic orders having magnitudes of 1.0, 0.9, 0.8, 0.9 respectively and all starting with zero-phase has been considered in both wide-band and narrow-band algorithms tests.
Primary and secondary paths generator applies at each simulation start an attenuation ATT_{dB} in dB given by the outdoor sound attenuation formula (3.27) for the computation of each transfer function magnitude, and considers a sound propagation speed of 340 m/s for the determination of the sound propagation delay occurring.

$$ATT_{dB} = 10 \log \left(\frac{Q_D}{(4\pi r^2)} \right) \tag{3.27}$$

Q_D is the directivity factor of the emitting sound source and r is the distance in meters that separates the emitting sound source from the receiver, i.e. one of the error microphones present in the environment.
In all the simulation tests conducted, a directivity factor of 4 and 2 has been assumed for the disturbance noise sound source and for all the anti-noise loudspeakers, respectively. For the determination of the distance between each emitter-receiver pair, the displacement diagrams shown in Fig. 3.15 and Fig. 3.16 was considered in all the single-channel and multi-channel algorithms tests, respectively. Simulation results are presented as follows: Section 3.5.1

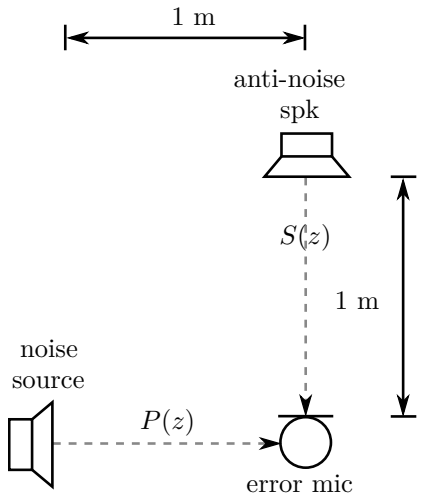


Figure 3.15: Simulator speakers and microphones displacement considered in single-channel systems simulations.

shows and compares the results obtained using Kalman-based and FxLMS-based single-channel wide-band systems, while Section 3.5.2 deals with their narrow-band counterparts. Section 3.5.3 deals with Kalman-based and FxLMS-based multi-channel wide-band systems simulation results and Section 3.5.4

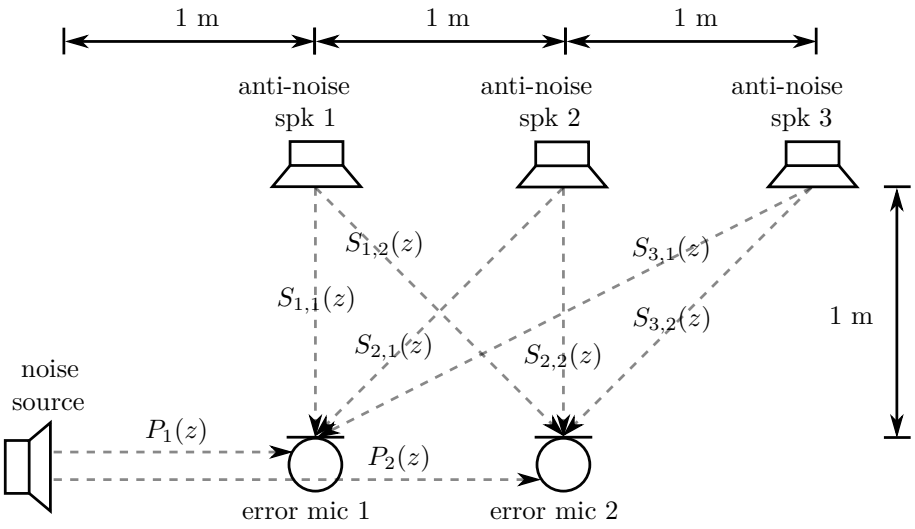


Figure 3.16: Simulator speakers and microphones displacement considered in multi-channel systems simulations.

with their relative narrow-band versions.

3.5.1 Single-Channel Wide-Band System Simulation

In this section the single-channel wide-band system described in Section 3.4.1 is tested. A wide-band zero-mean pink noise having unitary variance has been considered first to test both the Kalman-based and FxLMS-based solutions with an adaptive filter 128 taps long; subsequently, a tonal noise having the same characteristics as described in Section 3.5 opening as been considered too in order to test wide-band algorithm behaviors in presence of a narrow-band disturbance noise. Kalman algorithm tuning parameters used to produce all the presented results are the following:

- $A = I^{n \times n}$,
- $Q = 0.001 \cdot I^{(d \cdot n) \times (d \cdot n)}$,
- $R = 0.001 \cdot I^{m \times m}$.

where $I^{i \times i}$ is an $i \times i$ identity matrix and, for this configuration, $n = 128$, $d = 1$ and $m = 1$.
FxLMS algorithm tuning used, consisting substantially in its update step-size, is instead the following:

- $\mu = 0.02$.

Fig. 3.17 shows the results for disturbance pink noise case, while Fig. 3.18 shows the results for the disturbance tonal noise case.

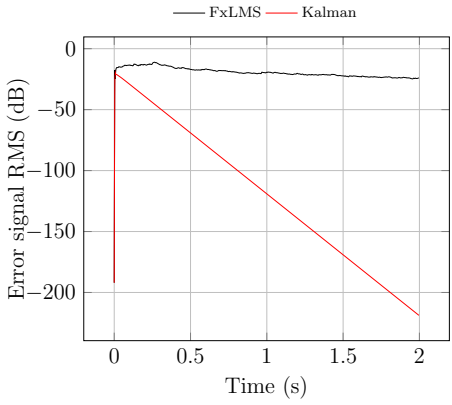


Figure 3.17: Wide-Band Single-Channel FxLMS Vs Kalman with pink noise.

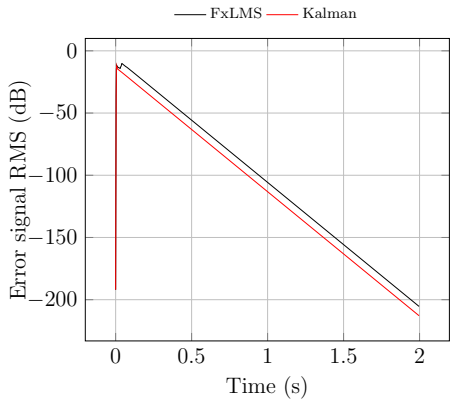


Figure 3.18: Wide-Band Single-Channel FxLMS Vs Kalman with tonal noise.

Advantages offered by Kalman approach over FxLMS, in terms of cancellation performance, are evident in the pink noise case; in the tonal noise case, even if the difference is less evident, Kalman approach exibits a better reaction time than its FxLMS counterpart.

3.5.2 Single-Channel Narrow-Band System Simulation

In this section, test results of the single-channel narrow-band system described in Section 3.4.2 are presented and compared with the results achieved by a

multi-tone FxLMS-based system operating in the same environmental conditions. A tonal noise having the same characteristics as described in Section 3.5 opening as been considered as the disturbance noise. Kalman algorithm tuning used is the following:

- $A = I^{n \times n}$,
- $Q = 0.001 \cdot I^{(d \cdot n) \times (d \cdot n)}$,
- $R = 0.00001 \cdot I^{m \times m}$.

where $I^{i \times i}$ is an $i \times i$ identity matrix, $n = 2n_h = 2 \cdot 4 = 8$, $d = 1$ and $m = 1$. FxLMS system tuning used is, instead, the following:

- $\mu = 0.9$.

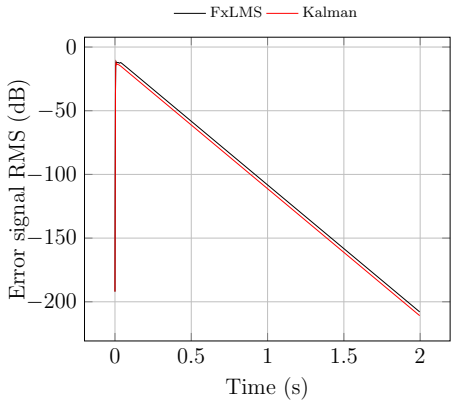


Figure 3.19: Narrow-Band Single-Channel FxLMS Vs Kalman.

Simulation results of Fig. 3.19 confirm again that Kalman algorithm present a better reaction time than its FxLMS counterpart.

3.5.3 Multi-Channel Wide-Band System Simulation

In this section the multi-channel wide-band algorithm of Section 3.4.3 with adaptive filters length of 128 taps each is tested with two different disturbance noise signals: with a zero mean, unitary variance pink noise, and with a tonal noise having the characteristics listed at the beginning of Section 3.5. Kalman algorithm tuning parameters values used during present tests are the following:

- $A = I^{n \times n}$,
- $Q = 0.001 \cdot I^{(d \cdot n) \times (d \cdot n)}$,

- $R = 0.001 \cdot I^{m \times m}$.

where $I^{i \times i}$ is an $i \times i$ identity matrix and, in the considered configuration, $d = 3$, $m = 2$ and $n = d \cdot 128 = 3 \cdot 128 = 384$.
Wide-band multi-channel FxLMS algorithm tuning used in the comparison is, instead, the following:

- $\mu = \begin{pmatrix} 0.015 & 0.003 \\ 0.003 & 0.015 \\ 0.0025 & 0.005 \end{pmatrix}$

where $\mu_{i,j}$ is the update step-size weighting actual sample of j^{th} microphone error signal in the computation formula of the i^{th} speaker anti-noise adaptive filter.
Test results obtained in the pink and tonal disturbance noise cases are respectively shown in Fig. 3.20 and Fig. 3.21 and remarks that also for a multi-channel ANC system, Kalman approach exhibits better performance than its multi-channel FxLMS counterpart.

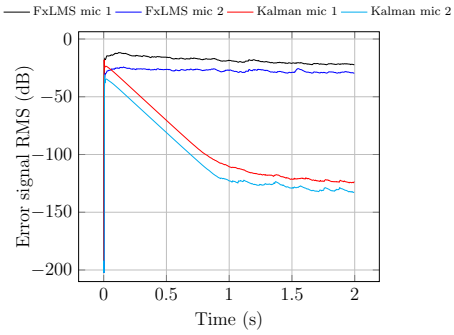


Figure 3.20: Wide-Band Multi-Channel FxLMS Vs Kalman with pink noise.

The advantage is again more evident when dealing with pink noise but it is appreciable for tonal noise as well.

3.5.4 Multi-Channel Narrow-Band System Simulation

In this section the narrow-band multi-channel Kalman approach introduced in Section 3.4.4 is tested with the same tonal noise used in all the previous tests, comparing its performance with those of a narrow-band multi-channel FxLMS-based system. Kalman algorithm tuning parameters are the following:

- $A = I^{n \times n}$,
- $Q = 0.001 \cdot I^{(d \cdot n) \times (d \cdot n)}$,

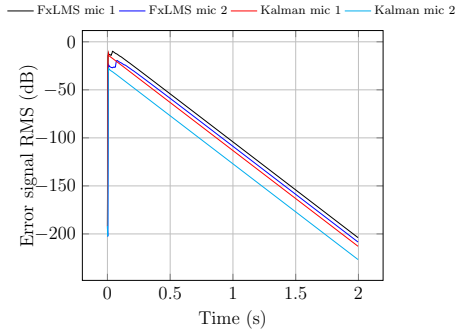


Figure 3.21: Wide-Band Multi-Channel FxLMS Vs Kalman with tonal noise.

- $R = 0.00001 \cdot I^{m \times m}$.

where $I^{i \times i}$ is an $i \times i$ identity matrix, $d = 3$, $m = 2$ and $n = d \cdot (2 \cdot n_h) = 3 \cdot (2 \cdot 4) = 24$.
FxLMS algorithm tuning is, instead, the following:

- $\mu = \begin{pmatrix} 0.6 & 0.1 \\ 0.1 & 0.6 \\ 0.3 & 0.5 \end{pmatrix}$

where, again, $\mu_{i,j}$ is the update step-size used in the generation of i^{th} speaker anti-noise signal from j^{th} error microphone signal contribution.

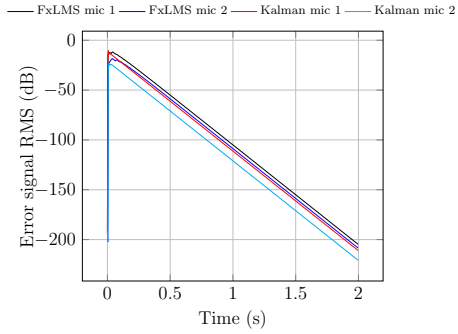


Figure 3.22: Narrow-Band Multi-Channel FxLMS Vs Kalman.

Results for narrow-band multi-channel systems of Fig. 3.22 show that, even if Kalman solution appears again to be more reactive than FxLMS, as in narrow-band single-channel test results presented in Section 3.5.2, both algorithms performance are quite close to each other.
However, especially when increasing the number of channels, FxLMS approach

shown an appreciable sensitiveness to tuning parameters, which have to be properly tuned for the specific situation. Kalman algorithm, instead, appeared to be quite insensitive from this point of view, showing a remarkable robustness in every condition.

3.6 ANC Practical Application

In this section, a practical ANC application is described, demonstrating the ability of these systems to improve quietness and enhance comfort in automotive environment.
Section 3.6.1 briefly described the implemented ANC system, while Section 3.6.2 presents some significant result obtained in real scenario.

3.6.1 Implemented System Overview

The ANC system realized in this work is based on the narrow-band FxLMS described in Section 3.1.2 and applied to a real car, a Peugeot 407 SW 2.0l, diesel of Fig. 3.23.

Figure 3.23: Peugeot 407 SW, 2.0l, diesel.

The main goal of this research activity was aimed to the realization of a marketable ANC system, where costs represents an obligation of primary importance. The choice of a narrow-band algorithm, rather then a broad-band one, comes from the limited computational resources available on the 32-bit fixed point DSP, which above consideration imposed us to use. Nevertheless,

as specified before, narrow-band algorithms fit quite well for applications where an engine noise is supposed to be managed, because of tonal nature that this latter shows.
Attempting to distributes ANC benefits over the whole environment, a multi-channel system have been realized, composed by 3 microphone and 4 loud-speakers, obtaining a total number of channels equal to 12. Microphones have been placed on the roof, in correspondence of the driver seat, the front passenger seat and the central rear passenger seat respectively. The 4 loudspeaker, instead, have been respectively placed within the 4 doors. Fig. 3.24 depicts previous component dislocation within the car.

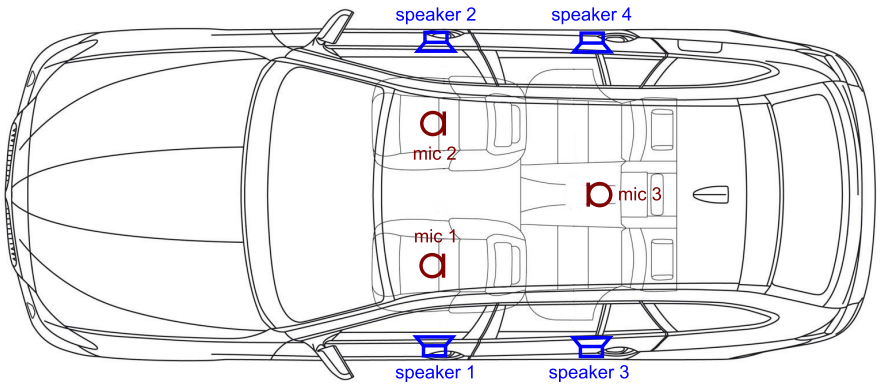


Figure 3.24: Error microphones and anti-noise loudspeakers dislocation within the cockpit.

In order to cope with varying noise frequencies, as engine RPM value change, two sub-modules have been developed:

- A fundamental frequency tracker.
- A harmonic manager.

The first one is needed to estimate the noise fundamental frequency, which is then used to compute the harmonic order frequency values on which focus the noise cancellation. Different approaches can be adopted for this purpose [49, 50], in this case the phonic-wheel signal from the car CAN-BUS has been exploited, which consists in a square wave with fundamental frequency multiple noise fundamental frequency.
The second one, instead, is a module that focuses the ANC cancellation on harmonic orders within a given range, where the overall cancellation effect is supposed to be more consistent or, at least, more appreciable by human beings,

and also saving computational resources for not trying to cancel not audible frequencies.

3.6.2 Real Environment Test Results

In this section, performances of system described in Section 3.6.1 are compared to those of a similar system having the same 4 loudspeaker but only the driver position microphone. The purpose of the test is to demonstrate the helpful contribution that microphones 2 and 3 give to the overall environment comfort. Even in the full microphones configuration, the algorithm was tuned to have best performances on error microphone 1 position (driver), For each test, the considered RPM range was $[900, 3300]$, explored with accelerating-decelerating ramps "manually" performed within a garage (in the first case) or empty roads (other cases).
Finally, also exploiting recorded signal from the engine phonic wheel, data have been properly processed in order to plot results in RPM versus acoustical cancellation charts.
Figures 3.25-3.30 show the total harmonic cancellation (which considers the cancellation on each disturbance tone instead of the broadband noise) obtained for the two algorithm at the three error position. In each chart, the solid line represents the actual recorded noise from related microphone, while the dashed one with "+" markers is the virtual noise reconstruction of ANC-off condition. Results demonstrate how the single-position algorithm improves comfort for driver position but degrades it for other ones. Multi-position algorithm, instead, improves comfort on driver position but also keeps noise on other ones under control, obtaining, in most cases, even an appreciable cancellation.

In order to analyze ANC system performances when road, wind and other kind of noise are introduced within the environment, also some test in real driving condition have been performed. Algorithm tuning was again focused on driver position but, once more, thanks to microphones 2 and 3, comfort on related position was not degraded and in most cases appreciable cancellations have been noticed. Three different driving condition have been tested:

1. Marching car on first gear.
2. Marching car on second gear.
3. Marching car on third gear.

Fig. 3.31, Fig. 3.32 and Fig. 3.33 show overall harmonic cancellation related to driver position in in the first, second and third condition respectively.

Obtained results demonstrate how the harmonic cancellation is effective in real driving condition as well.

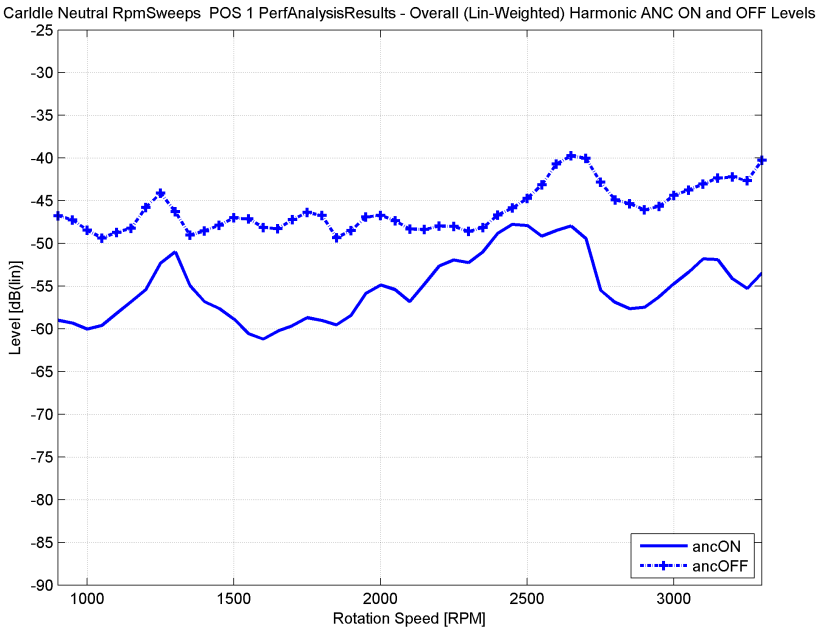


Figure 3.25: Error signals on position 1 (driver) for multi-position algorithm.

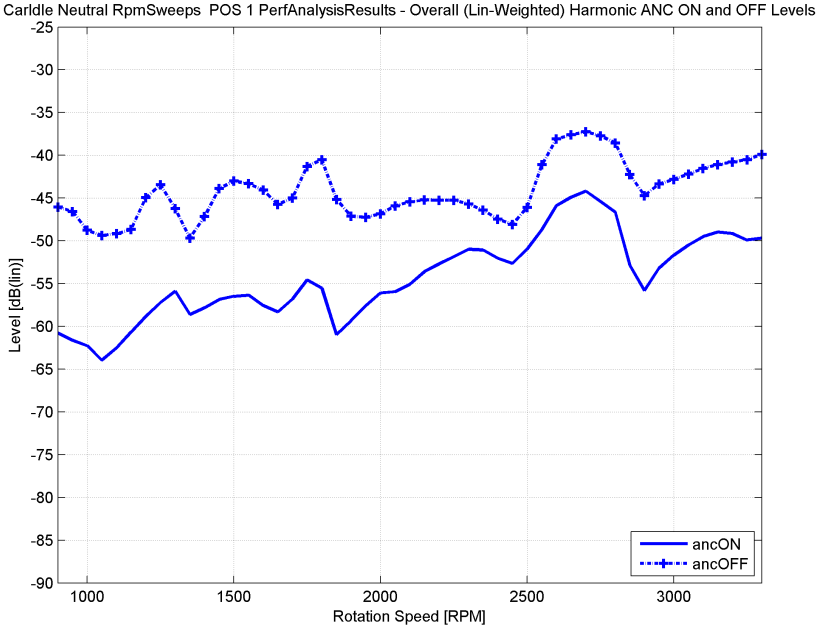


Figure 3.26: Error signals on position 1 (driver) for single-position algorithm.

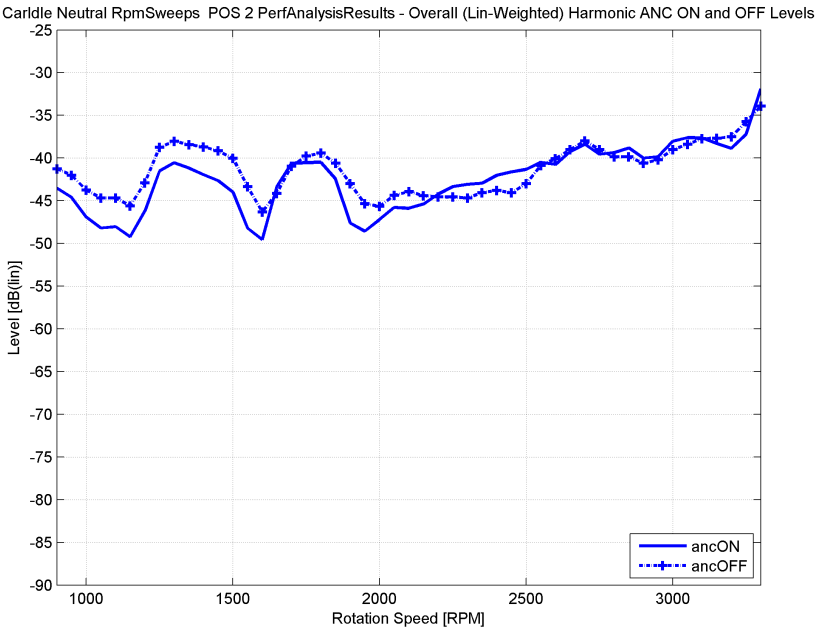


Figure 3.27: Error signals on position 2 (front passenger) for multi-position algorithm.

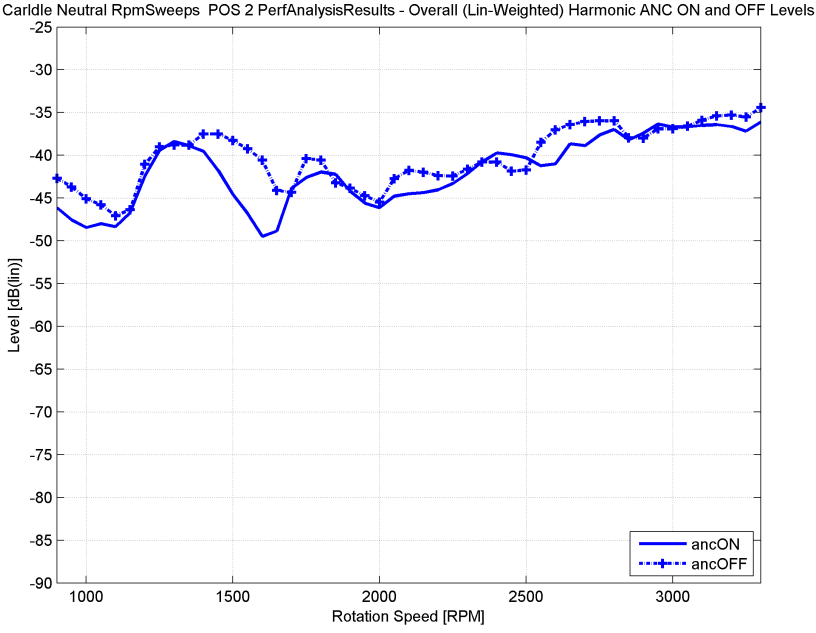


Figure 3.28: Error signals on position 2 (front passenger) for single-position algorithm.

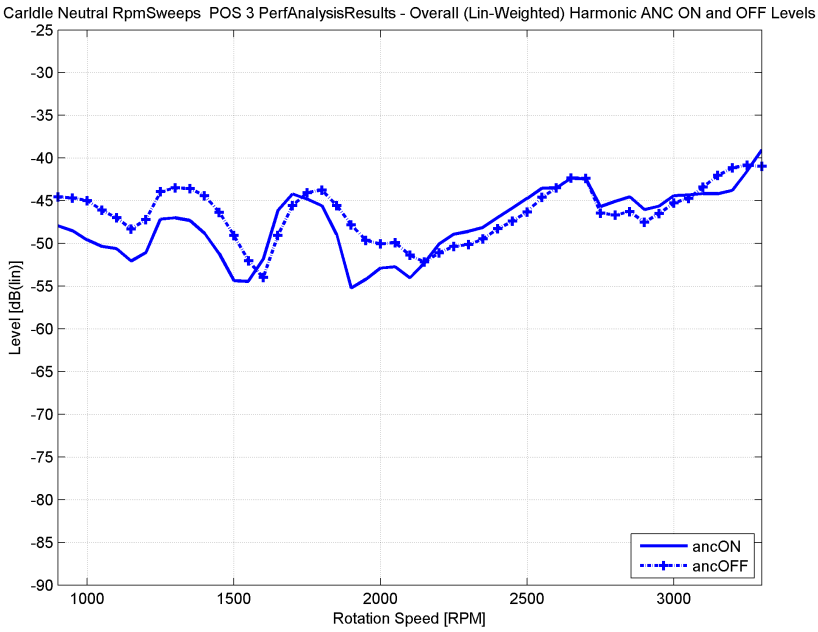


Figure 3.29: Error signals on position 3 (rear central passenger) for multi-position algorithm.

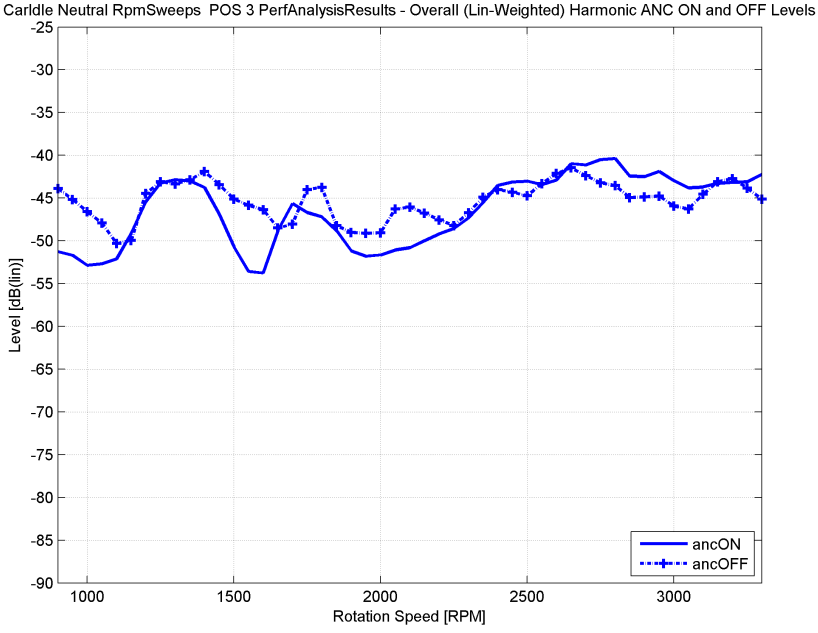


Figure 3.30: Error signals on position 3 (rear central passenger) for multi-position algorithm.

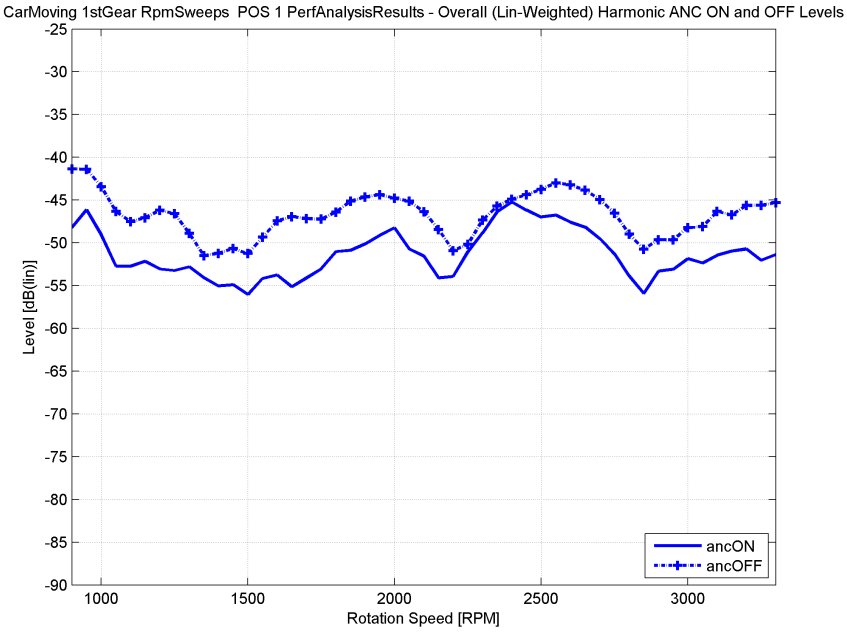


Figure 3.31: Error signals on position 1 when with car marching on first gear.

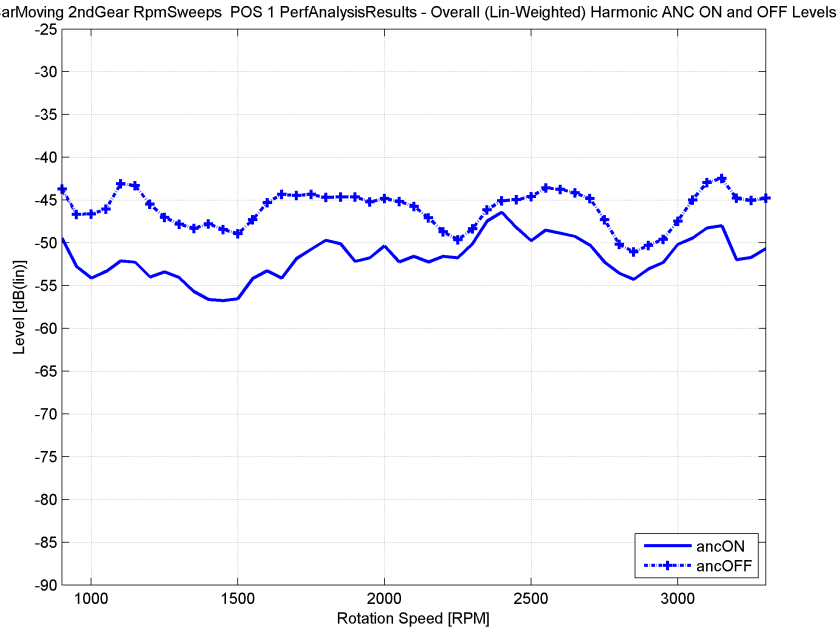


Figure 3.32: Error signals on position 1 when with car marching on second gear.

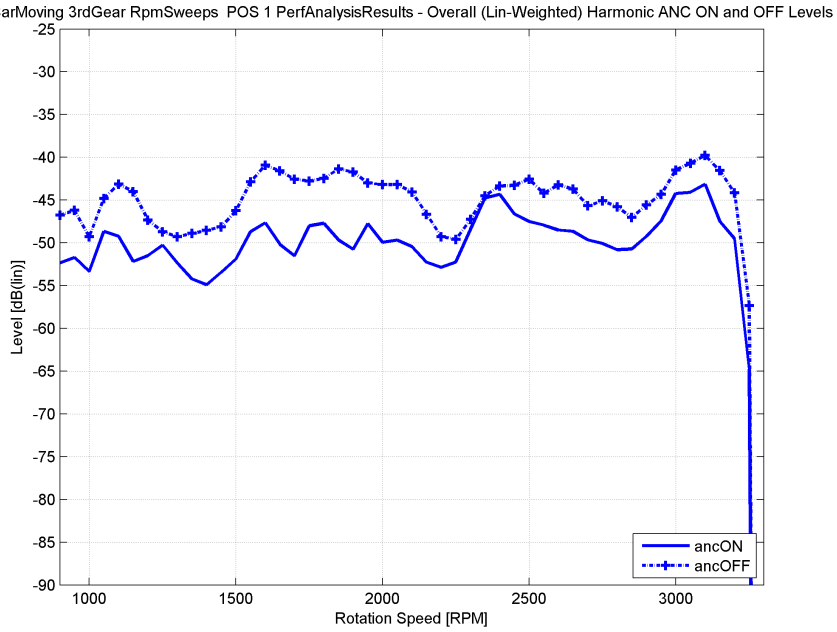


Figure 3.33: Error signals on position 1 when with car marching on third gear.

Chapter 4

Conclusions

In this work, active methods for comfort enhancement in automotive environments have been discussed. Two main topics have been handled:

- Canceling acoustic feedback and echo due to hearing aids located within the car cabin to improve communication experience among passengers.
- Actively canceling car cabin noise coming from the engine to improve quietness.

Both topics exploit adaptive filter theory to achieve their goals, adopting proper expedients for the specific applications.
In AFC, filter adaptation must handle the correlation between adaptive filter input signal and source speech signal itself. A performing de-correlation procedure in literature, the *PEM-AFROW* algorithm, has been explained and then integrated with an additional suppressor system, i.e. the *Suppressor PEM*, in Section 2.2.
In ANC, filter adaptation must take into account the secondary path, i.e. the acoustic coupling between anti-noise actuator and error sensor. Moreover, since noise in automotive is mainly of tonal nature, narrow-band approaches have been widely discussed in literature. A robust variant, the ACNB-FxLMS algorithm, has been presented in Section 3.2 and ANC solution exploiting Kalman filter theory have been investigated in Section 3.4.
In the following sections, specific concluding remarks for each treated subject are discussed.

4.1 AFC/AEC For Comfort Enhancement Concluding Remark

In Chapter 2, acoustic feedback and echo cancellation meant for automotive environments have been discussed.

An HW/SW test bench based on both a PC and an embedded system solution has been presented in Section 2.3. While the embedded DSP managed the

AFC algorithm, real automotive coupling were simulated on a PC with NU-Tech. Both room-modeling based algorithms, PEM-AFROW and Suppressor-PEM, has been implemented. They differ by the use (in the latter) of an additive suppressor filter which aims to increase the total system stability. In order to prevent the input signal clipping in the DSP OMAP-L138, an automatic gain controller has been developed. The normalization that it computes is adaptive, tracking the signal power and minimizing the distortion due to the hopping between different normalization factors in consecutive frames. Once the signal is elaborated, a de-normalization stage is used to keep the overall gain unchanged.
Several experimental tests have been performed, and obtained results have shown the effectiveness of proposed HW/SW solution for acoustic feedback cancellation in real acoustic conditions. Moreover, in automotive noise conditions, Suppressor-PEM approach shown an appreciable decrease of distortion and an overall improved robustness when moving the source, microphone and loudspeaker relative positions. This makes the aforementioned solutions interesting when dealing with rapidly changing impulse responses.
Based on *PEM-AFROW* and *Suppressor-PEM* algorithms previously developed for single channel scenario, a dual-channel intra-cabin communication system has been proposed and simulated in Matlab in Section 2.5. Due to the presence of two channels, more adaptive filters have been introduced in order to remove, in addition to the feedback, the echo effect as well. Therefore, six adaptive filters are included within the overall system:

1. One for the coupling between loudspeaker and microphone on channel 1 (AFC)

2. One for the coupling between loudspeaker and microphone on channel 2 (AFC)

3. One suppressor filter on channel 1

4. One suppressor filter on channel 2

5. One for the coupling between the loudspeaker on channel 2 and the microphone on channel 1 (AEC)

6. One for the coupling between the loudspeaker on channel 1 and the microphone on channel 2 (AEC)

The chosen adaptive algorithm is the *PB-FDAF* for both the AFC and AEC filters, which leads to fast convergence, low computational complexity and short execution time. The only difference between the two is the need in AFC of using whitened signals because of their correlation (de-correlation is performed in the

adaptive filtering circuit by the *PEM-AFROW*). The suppressor adaptive filters have been implemented by means of an *FDAF* and an *NLMS* algorithm. Two real VAD algorithms have been included in the algorithmic framework as well as the double talk detection algorithm which works on the basis of the VAD decisions.
Both solutions have been compared in computer simulations, resulting to have similar performances. Nevertheless, while the *NLMS* implementation is more effective in keeping the system stable when a critical condition occurs, the *FDAF* one, beside obtaining better misalignment for both the AFC and AEC filters, keeps the computational complexity lower, which is a primary feature for real-time applications. CPU usage when using latter one, indeed, decreased from more than 12% to less than 11%. The round-trip latency associated to the signal emitted from the front speaker and returning to his/her ears through the rear loudspeaker, can be estimated as follows:

$$\hat{\Delta}_{round-trip} = \delta_{speaker-mic} + \delta_{loud-mic} + T_f, \tag{4.1}$$

where $\delta_{speaker-mic}$ and $\delta_{loud-mic}$ are the delay introduced by the impulse responses in Fig. 2.16 and Fig. 2.17 and $T_f = (L_f - 1)/f_s$ the frame time for a frame length $L_f = 128$ and a sample frequency $f_s = 16\,\text{kHz}$. This gives:

$$\hat{\Delta}_{round-trip} = 1.63\,\text{ms} + 3.13\,\text{ms} + 7.94\,\text{ms} = 12.70\,\text{ms}, \tag{4.2}$$

which, is below the 20 ms delay threshold, above which the perception of one's own delayed voice becomes annoying. Similar considerations apply for the rear speaker case study. The latency contribution due to A/D and D/A process has not been included in this estimation as they heavily depend on the hardware used for the implementation. It must be said however that, in professional applications, this is expected to be sufficiently low to guarantee the satisfaction of the aforementioned latency requirements.
Results of executed tests have shown that the use of suppressor filters together with the *PEM-AFROW* algorithm leads to considerable improvements, allowing to achieve lower misalignment and higher gains, isolation between channels and feedback/echo cancellation.
As confirmed by intelligibility tests too, the use of suppressor filters is even more important when the system approaches instability: the SR BASELINE (not employing suppressor filters) is not able to work properly with gains superior than 4 dB, whereas the Suppressor-based solutions are able to guarantee MSG values higher than 10 dB (up to 12 dB for the SR w/ SUPPR. NLMS). The input SNR can also influence the performance. Indeed, the lower the actual signal level, the less perceptible the speech is, leading to bad feedback and echo estimates. However, proposed Speech Reinforcement system is able

to work properly also in presence of low SNRs as experimentally proved by computer simulations related to realistic in-car noisy acoustic environments.

4.2 ANC For Comfort Enhancement Concluding Remark

Chapter 3 described suitable techniques for actively cancel noise within a car cabin in order to increase comfort and quietness.

After an introduction about available solutions in literature, in Section 3.2 new FxLMS-based solution for narrow-band ANC application has been presented. Exploiting the fact that, in most cases, noise amplitude at error signal position is known a-priori and time invariant, proposed algorithm adapts only anti-noise signal phase while keeping its amplitude at a constant level. Tap-choosing logic monitors taps values and chooses, for each disturbance tone to be canceled, the lower one to be adapted and the greater one to be inferred. In this way it is assured an eventual sign inversion for the former while it is supposed that the latter one keeps its previous sign. This approach avoids instability and hardware saturation when the algorithm is unable to converge to a proper solution, because anti-noise output is kept bounded within its limits $[-A, A]$, which depend on given noise amplitude.
Experimental tests have proved that ACNB-FxLMS performances are comparable with classic NB-FxLMS solutions and that its robustness against bad secondary path estimation and generic instability has been considerably increased. Also mistakes in disturbance tones amplitudes does not compromise the proposed approach robustness, but only bounds the maxim achievable noise cancellation.
In Section 3.4, instead, one of the most used control-theory-based approach have been studied: the Kalman filter. Starting from the classic discrete time Kalman filter formulation of Section 1.2, four Kalman-based solutions for ANC applications has been described. From the most common single-channel wide-band Kalman for ANC of Section 3.4.1, a computationally convenient narrow-band approach has been derived in Section 3.4.2, exploiting a couple of orthogonal signal for each disturbance tone. At the same time, from the former, a multi-channel wide-band solution has been derived as well in Section 3.4.3, in order to extend the ANC system benefits on larger environments. In Section 3.4.4, single-channel narrow-band solution and multi-channel wide-band one have been combined together in order to obtain a computationally efficient multi-channel narrow-band system.
Several tests have been carried out in a simulation environment developed in *Matlab*, comparing the described Kalman-based approaches to the most com-

monly used FxLMS counterparts. Results, shown in Section 3.5, revealed that Kalman solutions allow to achieve better performances, outperforming FxLMS systems in every condition. Moreover, FxLMS algorithms shown an appreciable sensitiveness to tuning parameters and needed a specific tuning form case to case. Kalman approach, instead, appeared to be quite insensitive from this point of view, showing a considerably better robustness, along with a higher computational cost.
Finally, in Section 3.6, a practical ANC application of a multi-channel narrowband FxLMS-based algorithm is described. Various tests have been performed on a real car with diesel engine, both in steady state and marching condition. Results demonstrate the solution suitability and the helpful contribution that multiple microphones give to the overall sound quality, obtaining appreciable harmonic cancellation and distributing ANC benefits over the whole cockpit, without introducing acoustical artifacts.

List of Publications

[1] F. Faccenda, S. Squartini, E. Principi, L. Gabrielli, and F. Piazza "An Embedded-processor driven Test Bench for Acoustic Feedback Cancellation in real environments," 134th AES Convention, May 2013.

[2] F. Faccenda, S. Squartini, E. Principi, L. Gabrielli, and F. Piazza "A Real-time Dual-Channel Speech Reinforcement System for Intra-Cabin Communication," JAES Volume 61 Issue 11 pp. 889-910; November 2013

[3] F. Faccenda and L. Novarini "An Amplitude Constrained FxLMS Algorithm for Narrow-Band Active Noise Control Applications," 9th International Symposium on Image and Signal Processing and Analysis (ISPA 2015), September 2015.

[4] F. Faccenda and L. Novarini "Kalman Filter in Active Noise Control Applications," Submitted to Journal of Sound and Vibration.

Bibliography

[1] Gabriella Cerrato, "Automotive sound quality – powertrain, road and wind noise," *Sound and Vibration*, vol. 43, pp. 16–24, 2009.

[2] Simon Haykin, *Adaptive Filter Theory*, Prentice-Hall, Englewood Cliffs, NJ, 1991.

[3] V. Mathews and Z. Xie, "A stochastic gradient adaptive filter with gradient adaptive step size," *IEEE Transactions on Signal Processing*, vol. 41, pp. 2075–2087, 1993.

[4] A. I. Sulyman and A. Zerguine, "Convergence and steadystate analysis of a variable step-size nlms algorithm," *Signal Processing*, vol. 83, pp. 1255–1273, 2003.

[5] B.D.O. Anderson and J.B. Moore, *Optimal Filtering*, Prentice-Hall, Englewood Cliffs, NJ, 1979.

[6] Ming Zhang, Hui Lan, and Wee Ser, "Regeneration theory," *Bell System Technical Journal*, vol. 11, pp. 126–147, 1932.

[7] G. Franklin, J. Powell, A. Emami-Naeini, , and J. Powell, *Feedback control of dynamic systems*, vol. 3, Addison-Wesley Reading, 1994.

[8] T. van Waterschoot, *Design and evaluation of digital signal processing algorithms for acoustic feedback and echo cancellation*, Ph.D. thesis, Katholieke Universiteit Leuven, Leuven, Belgium, March 2009.

[9] T. van Waterschoot and M. Moonen, "Fifty years of acoustic feedback control: state of the art and future challenges," *Proceedings of the IEEE*, vol. 99, pp. 288–327, 2011.

[10] M. Guo, S. H. Jensen, and J. Jensen, "Evaluation of state-of-the-art acoustic feedback cancellation systems for hearing aids," *JAES*, vol. 61, pp. 125–137, 2013.

[11] J. Benesty, T. Gansler, D. Morgan, M. Sondhi, and S. Gay, "Advances in network and acoustic echo cancellation," *Springer*, 2001.

[12] J Benesty and Y Huang, "Adaptive signal processing: Applications to real-world problems," *Springer*, 2003.

[13] G. Rombouts, T. van Waterschoot, K. Struyve, and M. Moonen, "Acoustic feedback suppression for long acoustic paths using a nonstationary source model," *IEEE Trans. Signal Processing*, vol. 54, pp. 3426–3434, 2006.

[14] S. Cifani, R. Rotili, E. Principi, S. Squartini, and F. Piazza, "Real-time implementation of robust pem-afrow based solutions for acoustic feedback control," *AES Convention 127th*, 2009.

[15] S. Cifani, L. C. Montesi, R. Rotili, E. Principi, S. Squartini, and F. Piazza, "A pem-afrow based algorithm for acoustic feedback control in automotive speech reinforcement systems," *Proc. of 6th Int'l Symposium on Image and Sig. Proc. and Analysis*, 2009.

[16] A. Ortega, E. Lleida, and E. Masgrau, "Speech reinforcement system for car cabin communications," *IEEE Trans. Speech Audio Process*, vol. 13, pp. 917–929, 2005.

[17] G. Schmidt and T. Haulick, "Signal processing for in-car communication systems," *Elsevier Signal Processing*, vol. 86, pp. 1307–1326, 2005.

[18] C. Luke, H. Ozer, G. Schmidt, A. TheiB, and J. Withopf, "Signal processing for in-car communication systems," *5th Biennial Workshop on DSP for In-Vehicle Systems*, 2011.

[19] A. Spriet, I. Proudler, M. Moonen, M., and J. Wouters, "Adaptive feedback cancellation in hearing aids with linear prediction of the desired signal," *IEEE Trans on Signal Processing*, vol. 53, pp. 3749–3763, 2005.

[20] U. Forssell and L. Ljung, "Closed-loop identification revisited," *Automatica-Elsevier Science Ltd*, vol. 35, pp. 1215–1241, 1999.

[21] J.J. Shynk, "Frequency-domain and multirate adaptive filtering," *IEEE Signal Process*, p. 14–37, 1992.

[22] G. Rombouts, T. van Waterschoot, and M. Moonen, "Proactive notch filtering for acoustic feedback cancellation," *Proc. of IEEE Benelux/DSP Valley Sig. Proc. Symp*, pp. 169–172, 2006.

[23] G. Rombouts, T. van Waterschoot, and M. Moonen, "Robust and efficient implementation of the PEM-AFROW algorithm for acoustic feedback cancellation," *JAES*, vol. 55, pp. 955–966, 2007.

[24] J. Ramirez, J.C. Segura, M.C. Benitez, A. De La Torre, and A. Rubio, "Efficient voice activity detection algorithms using long-term speech information," *Speech Communication*, vol. 42, pp. 271–287, 2004.

[25] F. Eyben, F. Weninger, S. Squartini, , and B. Schuller, "Real-life voice activity detection with lstm recurrent neural networks and application to holliwood movies," *Proceedings of IEEE ICASSP 2013*, 2013.

[26] P. C. Loizou, *Speech Enhancement: Theory and Practice (Signal Processing and Communications)*, CRC Press, 2007.

[27] L. Lamel, R. Kassel, and S. Seneff, "Speech database development: Design and analysis of the acoustic-phonetic corpus," *Proc. DARPA Speech Recognition Workshop*, pp. 121–124, 1986.

[28] F. Piazza, S. Cecchi, L. Palestini, P. Peretti, and S. Squartini, "Advanced cis architecture and algorithms for enhanced in-car audio listening," *ICNSC 2009*, 2009.

[29] S Cecchi, A Primavera, F Piazza, F Bettarelli, E Ciavattini, R Toppi, JGF Coutinho, W Luk, C Pilato, F Ferrandi, VM Sima, and K Bertels, "The hartes carlab: A new approach to advanced algorithms development for automotive audio," *JAES*, vol. 59, pp. 858–869, 2011.

[30] J. Borish and J. B. Angell, "An efficient algorithm for measuring the impulse response using pseudorandom noise," *JAES*, vol. 31, pp. 478–488, 1983.

[31] F. Piazza, S. Squartini, R. Toppi, M. Navarri, M. Pontillo, F. Bettarelli, and A. Lattanzi, "Industry-oriented software-based system for quality evaluation of vehicle audio environments," *IEEE Transactions on Industrial Electronics*, vol. 53, pp. 855,866, 2006.

[32] J. Ma, Y. Hu, and P. C. Loizou, "Objective measures for predicting speech intelligibility in noisy conditions based on new band-importance functions," *Journal of the Acoustical Society of America*, 2009.

[33] B. Angelini, F. Brugnara, D. Falavigna, D. Giuliani, R. Gretter, and M. Omologo, "Automatic segmentation and labeling of english and italian speech databases," *Proc. of Eurospeech*, p. 653–656, 1993.

[34] N. Samardzic and C. Novak, "Sound source signal parameters in vehicles for determining speech transmission index," *JAES*, vol. 59, pp. 735–744, 2011.

[35] N. Samardzic and C. Novak, "The analysis of the reduction in vehicle speech intelligibility for normal hearing and hearing impaired individuals in a simulated driving environment with contributions from the ordered and masking noise source," *JAES*, vol. 61, pp. 676–687, 2013.

[36] K. Eneman and M. Moonen, "Iterated partitioned block frequency-domain adaptive filtering for acoustic echo cancellation," *IEEE Transactions on Speech and Audio Processing*, vol. 11, pp. 143–158, 2003.

[37] S.M. Kuo and D.R. Morgan, "Active noise control: a tutorial review," *Proceedings of the IEEE*, vol. 87, pp. 943 – 973, 1999.

[38] C. H. Hansen, *Understanding Active Noise Cancellation*, Taylor & Francis, 2002.

[39] J. Poshtan, F. Taring, A. Nasiri, and M. H. Kahaie, "Multichannel active noise control algorithm using align matrix," *Proceedings of 2003 IEEE Conference*, 2003.

[40] O. J. Tobias and R. Seara, "Leaky-fxlms algorithm: Stochastic analysis for gaussian data and secondary path modeling error," *IEEE Transaction on Speech and Audio Processing*, vol. 13, 2005.

[41] Jr. J. R. Glover, "Adaptive noise canceling applied to sinusoidal interferences," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-25, pp. 484–491, 1977.

[42] S. J. Elliott and P. Darlington, "Adaptive cancellation of periodic, synchronously sampled interference," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 715–717, 1985.

[43] B. Chaplin, "The cancellation of repetitive noise and vibration," *Proc. Inter-noise*, p. 699–702, 1980.

[44] S. Kim and Y. Park, "Active control of multi-tonal noise with reference generator based on on-line frequency estimation," *Journal of Sound and Vibration*, pp. 647–666, 1999.

[45] D. R. Morgan and C. Sanford, "A control theory approach to the stability and transient analysis of the filtered-x lms adaptive notch filter," *IEEE Trans. Signal Processing*, vol. 40, pp. 2341–2346, 1992.

[46] Tak Keung Yeung and Sze Fong Yau, "Feedforward anc system using adaptive fir filters with on-line secondary path identification," *Electronics, Circuits and Systems, 1999. Proceedings of ICECS '99. The 6th IEEE International Conference*, vol. 2, pp. 675 – 678, 1999.

[47] Ming Zhang, Hui Lan, and Wee Ser, "On comparison of online secondary path modeling methods with auxiliary noise," *Speech and Audio Processing, IEEE Transactions*, vol. 13, pp. 618 – 628, 2005.

[48] P. A. C. Lopes and M. S. Piedade, "A kalman filter approach to active noise control," *Signal Processing Conference, 2000 10th European*, pp. 1 – 4, 2000.

[49] Eric Larsen and Ronald M. Aarts, *Audio bandwidth extensionapplication of psychoacoustics, signal processing and loudspeaker design*, John Wiley & Sons, 2004.

[50] Junho Lee, Eunjung Song, Youngcheol Park, and Daehee Youn, "Effective bass enhancement using second-order adaptive notch filter," *IEEE Transactions on Consumer Electronics*, vol. 54, pp. 663 – 668, 2008.

[51] Alberto Bellini, Eraldo Carpanoni, Giacomo Frassi, Monica Tesauri, and Emanuele Ugolotti, "Design of a dsp-based 24 bit digital audio equalizer for automotive applications," *IEEE*, vol. 1, pp. 293–296, 2002.

[52] J. Cheer, S. J. Elliott, and M. F. S. Gálvez, "Design and implementation of a car cabin personal audio system," *JAES*, vol. 61, pp. 412–424, 2013.

[53] S. Cifani, E. Principi, R. Rotili, F. Piazza, and S. Squartini, "Real-time simulation for acoustic feedback cancellation algorithms: an hybrid pc/c6713-dsk based implementation," *Education and Research Conference*, pp. 37–41, 2010.

[54] F. Faccenda, S. Squartini, E. Principi, L. Gabrielli, and F. Piazza, "An embedded-processor driven test bench for acoustic feedback cancellation in real environments," *AES 134th Convention*, 2013.

[55] F. Faccenda, S. Squartini, E. Principi, L. Gabrielli, and F. Piazza, "A real-time dual-channel speech reinforcement system for intra-cabin communication," *JAES*, vol. 61, pp. 889–910, 2013.

[56] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE Trans. on Acoustics, Speech and Signal Proc*, vol. 33, 1985.

[57] I. Cohen, "Noise spectrum estimation in adverse environments: improved minima controlled recursive averaging," *IEEE Trans. on Speech and Audio Proc*, vol. 11, 2003.

[58] A. Bastari, S. Squartini, and F. Piazza, "Joint acoustic feedback cancellation and noise reduction within the prediction error method framework,"

Proc. of IEEE Hands-free Speech Communication and Microphone Arrays (HSCMA), 2008.

[59] J.B. Allen and A.D. Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, pp. 943–950, 1979.

[60] S.Squartini, E. Ciavattini, A. Lattanzi, D. Zallocco, F. Bettarelli, and F. Piazza, "Nu-tech: implementing dsp algorithms in a plug-in based software platform for real time audio applications," *AES 118th Convention*, 2005.

[61] A. Spriet, G. Rombouts, M. Moonen, M., and J. Wouters, "Adaptive feedback cancellation in hearing aids," *Journal of the Franklin Institute*, vol. 343, pp. 545–573, 2006.

[62] Jr E. Ziegler, "Selective active cancellation system for repetitive phenomena," *U.S. Patent*, 1989.

[63] D. R. Morgan and J. Thi, "A multitone pseudocascade filtered-x lms adaptive notch filter," *IEEE Trans. Signal Processing*, vol. 41, pp. 946–956, 1993.

[64] S. M. Kuo and M. Ji, "Passband disturbance reduction in periodic active noise control systems," *IEEE Trans. Speech Audio Processing*, vol. 4, pp. 96–103, 1996.

[65] S. M. Kuo, S. Zhu, and M. Wang, "Development of optimum adaptive notch filter for fixed-point implementation in active noise control," *Proc. 1992 Int. Conf. Industrial Electronics*, pp. 1376–1378, 1992.

[66] Francesco Faccenda and Luca Novarini, "An amplitude constrained fxlms algorithm for narrow-band active noise control applications," *ISPA 2015*, 2015.

[67] S.J. Elliott and P.A. Nelson, "The active control of sound," *Electronics & Communication Engineering Journal*, vol. 2, pp. 127 – 136, 1990.

[68] S.S. Haykin, *Kalman Filtering And Neural Networks*, Wiley Online Library, 2001.

[69] G. Welch and G. Bishop, "An introduction to the kalman filter: Siggraph 2001 course 8," 2001.

[70] K. Dejhan, S. Yimniun, A. Trirat, and F. Cheevasuvit, "Tms 320c31-based narrow-band noise rejection kalman filter implementation," *IEEE ICIT'02*, vol. 1, pp. 332 – 336, 2002.

[71] Hen-Geul Yeh, "Real-time implementation of a narrow-band kalman filter with a floating-point processor dsp32," *Industrial Electronics, IEEE Transactions*, vol. 37, pp. 13 – 18, 1990.

[72] Mohinder S. Grewal and Angus P. Andrews, *Kalman filtering : theory and practice using MATLAB*, Wiley, New York, Chicester, Weinheim, 2001, La p. de t. porte en plus : A Wiley-Interscience publication.

[73] Fan Wang, "Robust kalman filters for linear time-varying systems with stochastic parametric uncertainties," *Signal Processing, IEEE Transactions*, vol. 50, pp. 803 – 813, 2002.

[74] Eli Brookner, *Tracking and Kalman Filtering Made Easy*, John Wiley & Sons, 1998.

[75] Francesco Faccenda and Luca Novarini, "Kalman filter in active noise control applications," *Journal of Sound and Vibration*, 2016, Processing for publication.