

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي  
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البلدية  
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا  
Faculté de Technologie

قسم الإلكترونيك  
Département d'Électronique



## Mémoire de Projet de Fin d'Études

présenté par

Bouzidi Mohammed

&

Yahiaoui Chouaib

pour l'obtention du diplôme de Master en Électronique Spécialité Traitement de  
l'Information et Systèmes Électroniques (TISE)

---

Thème

---

# Implémentation d'un contrôleur actif du bruit sur une Raspberry Pi

## 3

---

Proposé par : Ykhlef Farid

Année Universitaire 2016-2017

## Remerciements

---

Tous d'abord on remercie DIEU le tout puissant qui nous a donnés durant toutes ces années la santé, le courage et la fois en nous même pour arriver à ce jour.

Nous tenons à exprimer notre profonde gratitude et reconnaissance à notre encadreur Dr YKHLEF FARID, pour ses qualités scientifiques et humaines, qui nous ont beaucoup aidées et guidées tout au long de ce travail, et pour le temps et la patience qui nous a accordé.

Nous adressons aussi nos remerciements à l'ensemble du jury qui a accepté de juger ce travail.

Enfin, un grand merci plein de reconnaissance, à nos parents pour leurs soutiens et encouragements et pour qui ce mémoire est dédié.

### ملخص:

أدى الانتشار الهائل للصناعة إلى ضرورة استعمال أنظمة ANC، تناولت أطروحة الماستر هذه صناعة هذا النظام باستخدام RASPBERRY PI، فضلنا استعمال Simulink لأنه يوفر المحاكاة. مراقبة أنظمة الاتصالات ومعالجة الإشارات، بعد محاكاة الخوارزميات باستخدام شفرة MATLAB اخترنا استخدام خوارزمية FXLMS. وقد تم إنشاء تطبيق أندرويد لتسهيل استخدام وتعبئة هذا النظام.

في حالة المحاكاة هذا النظام يقلل من الضوضاء إلى أقل من 40 dB، مقارنة مع العمل يتم الحصول على نتيجة أقل جيدة بسبب نوعية المواد المستخدمة.

كلمات المفاتيح: خوارزمية FXLMS ;

---

**Résumé :** Les systèmes ANC sont de plus en plus nécessaires et cela est dû à la propagation massive de l'industrie. Dans ce travail de master nous reproduisons un système ANC à l'aide d'un nano-ordinateur mono-carte à processeur ARM nommé Raspberry Pi. L'outil SIMULINK de MATLAB a été aussi utilisé afin d'accomplir l'interfaçage, car ce dernier fournit la simulation, l'implémentation et le contrôle des systèmes de communication et le traitement de signal. Après la simulation de plusieurs algorithmes pour ANC sous MATLAB nous avons opté pour l'algorithme adaptatif de type FXLMS. Une application ANDROID a été conçue pour faciliter l'utilisation et la mobilisation de notre système. Dans ce travail, on a conclu que dans le cas de la simulation, le système permet de diminuer le bruit jusqu'à 40 dB de moins rapport aux résultats pratiques.

**Mots clés :** Système ANC ; Raspberry PI ; SIMULINK ; MATLAB ; algorithme FXLMS.

---

### Abstract :

ANC systems are increasingly needed and this is due to the massive spread of the industry community. In this master we reproduce an ANC system using an ARM processor-based nano-computer called Raspberry Pi. MATLAB's SIMULINK tool was also used to accomplish interfacing because it provides simulation, implementation and control of communication systems and signal processing. After simulation of several algorithms for ANC using the MATLAB language we opted for the FXLMS adaptive algorithm type. An ANDROID application has been designed to facilitate the use and mobilization of our system. In this work, it was concluded that in the simulation case, the system reduces the noise to 40 dB less than practical case.

**Keywords :** ANC system ; Raspberry PI ; SIMULINK ; MATLAB ; FXLMS algorithm.

---

## Listes des acronymes et abréviations

---

**ANC** : Active Noise Control

**dB** : décibel

**FX-LMS** : Filtered-X Least-Mean-Square

**FX-NLMS** : Filtered-X Normalized Least-Mean-Square

**LLMS** : Leaky Least Mean Squares

**LMS** : Least Mean Squares

**NLMS** : Normalized least mean squares

**PC** : Personal Computer

**RIF** : Réponse Impulsionnelle Finie

**RII** : Réponse Impulsionnelle Infinie

**SPL** : Sound Pressure Level

**SSH** : Secure Shell

**SSP** : SIMULINK Support Package

**SWL** : Sound Power Level

**USB** : Bus Universel en Série

# Table des matières

<b>CHAPITRE 1</b>	<b>ETAT DE L'ART</b>	<b>1</b>
1.1	VUE D'ENSEMBLE	1
1.2	CONCEPT DU CONTROLE ACTIF DU BRUIT	3
1.3	MOTIVATION	4
<b>CHAPITRE 2</b>	<b>NOTIONS ACOUSTIQUES</b>	<b>5</b>
2.1	INTRODUCTION	5
2.2	ACOUSTIQUE	5
2.2.1	<i>Son</i>	5
2.2.2	<i>Ondes sonores</i>	5
2.2.3	<i>Vitesse de son</i>	6
2.2.4	<i>Caractéristique de son</i>	6
2.2.5	<i>Vitesse vibratoire</i>	7
2.2.6	<i>Pression sonore (acoustique)</i>	7
2.2.7	<i>Intensité sonore</i>	8
2.2.8	<i>Impédance acoustique</i>	8
2.2.9	<i>Niveau de pression sonore</i>	8
2.2.10	<i>Niveau de puissance sonore</i>	8
<b>CHAPITRE 3</b>	<b>CONTROLE ACTIF DU BRUIT</b>	<b>11</b>
3.1	INTRODUCTION	11
3.2	POURQUOI LE CONTROLE ACTIF DU BRUIT ?	11
3.2.1	<i>Composantes de système ANC</i>	12
3.2.2	<i>Applications du système ANC</i>	15
3.3	SYSTEMES DU CONTROLE ACTIF DU BRUIT	16
3.3.1	<i>Feed-forward contrôle actif du bruit</i>	17
3.3.2	<i>Feed-back contrôle actif du bruit</i>	19
<b>CHAPITRE 4</b>	<b>ALGORITHMES ADAPTATIFS POUR LE CONTROLE ACTIF DU BRUIT</b>	<b>20</b>
4.1	INTRODUCTION	20
4.2	ALGORITHMES ADAPTATIFS	20
4.2.1	<i>Algorithme LMS</i>	22
4.2.2	<i>Algorithme LMS normalisé (NLMS)</i>	23
4.2.3	<i>Algorithme Leaky LMS (LLMS)</i>	24
4.2.4	<i>Algorithme Filtré-x LMS</i>	24
4.2.5	<i>Filtré-X-NLMS</i>	29
4.2.6	<i>Algorithme Leaky FX-LMS</i>	29
4.3	CHOIX DE L'ALGORITHME	30
4.4	TESTS DES ALGORITHMES	30
4.4.1	<i>FX-LMS</i>	30
4.4.2	<i>FX-NLMS</i>	32
4.5	CONCLUSION	34
<b>CHAPITRE 5</b>	<b>IMPLEMENTATION SUR RASPBERRY PI</b>	<b>35</b>
5.1	INTRODUCTION	35
5.2	MATERIEL UTILISE	35
5.2.1	<i>Software</i>	35

5.2.2	<i>Hardware</i>	35
5.3	CONCEPTION ET IMPLEMENTATION	38
5.3.1	<i>Configuration de la connexion</i>	38
5.3.2	<i>Programmations de l'ANC</i>	40
5.4	ARCHITECTURE COMPLETE	45
5.5	L'APPLICATION ANDROID	47
5.5.1	<i>SSH</i>	47
5.5.2	<i>Notre application Android</i>	47
<b>CONCLUSION GENERALE</b>		<b>52</b>
<b>ANNEXE</b>		<b>53</b>
A.1	INTRODUCTION A MATLAB	53
A.2	INTRODUCTION A SIMULINK	55
A.3	INTRODUCTION A ANDROID STUDIO	57

## Liste des figures

<b>Figure 1.1</b> : Concept d'annulation de bruit actif [2].....	2
<b>Figure 1.2</b> : Concept du contrôle du bruit actif dans un canal présenté par P. Lueg [3]. .....	3
<b>Figure 1.3</b> : Contrôle actif du bruit basé sur le ANC feedback [3]. .....	3
<b>Figure 3.1</b> : Microphone électrodynamique.....	12
<b>Figure 3.2</b> : Microphone électrostatique. ....	13
<b>Figure 3.3</b> : Microphone électrostatique a électret.....	13
<b>Figure 3.4</b> : Classification des microphones. ....	15
<b>Figure 3.5</b> : Système ANC Feed-forward à large bande dans un canal [2]. ....	18
<b>Figure 3.6</b> : Système ANC Feed-forward à bande étroite [2]. ....	19
<b>Figure 3.7</b> : Système ANC Feed-back .....	19
<b>Figure 4.1</b> : Structure de filtrage adaptatif.....	21
<b>Figure 4.2</b> : Bloc diagramme de l'algorithme LMS. ....	22
<b>Figure 4.3</b> : Modélisation des chemins dans un système ANC. ....	24
<b>Figure 4.4</b> : Bloc diagramme de l'algorithme LMS dans l'ANC. ....	25
<b>Figure 4.5</b> : Système ANC implémente avec l'algorithme FX-LMS. ....	26
<b>Figure 4.6</b> : Modélisation de la configuration hors-ligne. ....	27
<b>Figure 4.7</b> : Bloc diagramme de l'algorithme FX-LMS avec le système ANC. ....	28
<b>Figure 4.8</b> : Convergence de l'erreur de filtrage de l'algorithme FX-LMS pour différents pas d'adaptation. ....	31
<b>Figure 4.9</b> : Simulation de FX-LMS avec le bruit d'un ventilateur. ....	31
<b>Figure 4.10</b> : Simulation de FX-LMS avec le bruit mixé, $\mu=0.01$ . ....	32
<b>Figure 4.11</b> : Convergence de l'erreur de filtrage de l'algorithme FX-NLMS pour différents pas d'adaptation. ....	32
<b>Figure 4.12</b> : Simulation de FXNLMS avec le bruit d'un ventilateur. ....	33
<b>Figure 4.13</b> : Simulation de FX-NLMS avec le bruit mixé, $\mu=0.19$ . ....	33
<b>Figure 5.1</b> : Raspberry PI 3. ....	36
<b>Figure 5.2</b> : Description des en-têtes GPIO. ....	37
<b>Figure 5.3</b> : Configuration de la connexion sous MATLAB. ....	39
<b>Figure 5.4</b> : Add-On Explorer. ....	39
<b>Figure 5.5</b> : Terminal Raspberry PI.....	41
<b>Figure 5.6</b> : Command pour accéder au fichier alsa.conf. ....	41
<b>Figure 5.7</b> : Aperçu sur le fichier alsa.conf. ....	42
<b>Figure 5.8</b> : Schéma SIMULINK pour l'estimation de chemin secondaire .....	43
<b>Figure 5.9</b> : Estimation. ....	43
<b>Figure 5.10</b> : Configuration des paramètres.....	44
<b>Figure 5.11</b> : Barre de tâches.....	44
<b>Figure 5.12</b> : Schéma SIMULINK de l'ANC. ....	45
<b>Figure 5.13</b> : Réalisation de système ANC feed-back avec le PC. ....	46
<b>Figure 5.14</b> : Réalisation de système ANC sans le PC.....	46
<b>Figure 5.15</b> : Première interface de l'application (activity_spalsh_screen) .....	48
<b>Figure 5.16</b> : Deuxième interface (activity_main). ....	49
<b>Figure 5.17</b> : Troisième interface (activity_configuration).....	50

<b>Figure 5.18</b> : Quatrième interface (activity_choix).....	51
<b>Figure A.1</b> : Environnement MATLAB .....	54
<b>Figure A.2</b> : Fenêtre SIMULINK.....	55
<b>Figure A.3</b> : Navigateur de la librairie SIMULINK.....	56
<b>Figure A.4</b> : Android Studio.....	58
<b>Figure A.5</b> : Programmation de l'interface.....	58



## Liste des tableaux

---

<b>Tableau 2.1</b> : Valeurs typiques du niveau sonore [8]. .....	9
<b>Tableau 2.2</b> : Puissance de différentes sources sonores [8]. .....	10
<b>Tableau 3.1</b> : Comparaison entre les microphones selon leur type. ....	14
<b>Tableau 5.1</b> : Caractéristiques techniques. ....	36

# Chapitre 1 Etat de l'art

---

## 1.1 Vue d'ensemble

Notre monde devient de plus en plus technologiquement avancé. Bien que les progrès technologiques aient apporté un réconfort important à notre vie, ils pourraient, d'autre part, introduire de nouveaux défis qui influent négativement sur notre environnement. Par exemple, le bruit acoustique dans notre environnement, il est aujourd'hui courant en raison de produits tels que les compresseurs, les aspirateurs, les lave-vaisselles, les systèmes de ventilation, les transformateurs, les moteurs, etc. en plus, la vibration mécanique associée est également un problème dans les industries, le transport et dans les appareils ménagers [1].

Dans notre environnement, le bruit acoustique peut être classé comme un bruit à large bande ou à bande étroite. Le bruit à large bande (ou bande élargie) causé par la turbulence est totalement aléatoire. Ce bruit turbulent distribue son énergie uniformément sur les bandes de fréquences (e. g. avions à réaction et bruit d'une explosion, etc.). Le bruit à bande étroite concentre la majeure partie de son énergie à des fréquences spécifiques. Ce type de bruit est lié aux machines rotatives ou répétitives, de sorte qu'il est périodique ou presque périodique (e. g. bruit des moteurs à combustion interne dans le transport, compresseurs, bruit du ventilateur [2], etc.).

L'exposition à des bruits acoustiques peut être ennuyeuse et dangereuse en même temps pour l'être humain du point de vue physique et psychologique, et cela surtout pour des gens qui travaillent dans le secteur industriel.

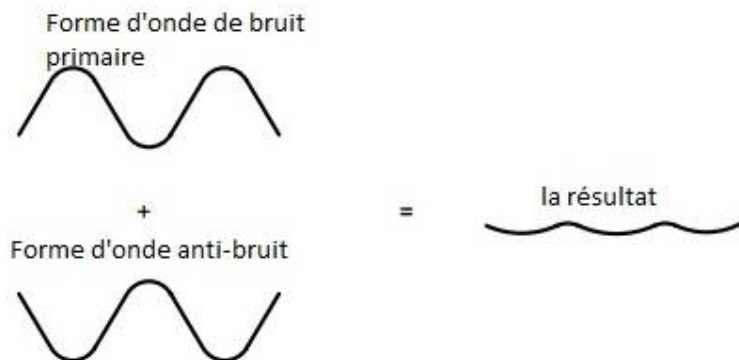
De nombreuses techniques ont été développées pour réduire ou contrôler le bruit acoustique [1]. Le choix d'une telle technique est déterminé par les caractéristiques du

bruit. Les approches passives sont constituées d'absorbeurs et/ou de réflecteurs/barrières. Les absorbeurs convertissent l'énergie acoustique en énergie thermique, mais les réflecteurs/barrières empêchent le bruit d'entrer dans un espace à partir d'un autre espace. Ils reflètent le champ des ondes incidentes [3].

En pratique la méthode passive réduit bien le bruit quand le taux de fréquence est presque 500 Hz, par contre elle n'est pas pratique dans les fréquences plus basses à cause des équipements massives [3].

Pour résoudre ces problèmes, un intérêt considérable a été démontré au cours de la dernière décennie par l'application de la méthode basée sur le contrôle actif du bruit. L'annulation active du bruit est une technique pratique utilisée pour réduire le son indésirable [2]. Le principe du contrôle actif du bruit est basé sur la superposition d'ondes sonores. Les sources secondaires produisent un «anti-bruit» d'amplitude égale et de phase opposée. Le son indésirable (le bruit primaire) et l'anti-bruit sont donc combinés acoustiquement ce qui résulte une annulation des deux sons.

La figure 1.1 montre les formes d'onde du bruit indésirable (le bruit primaire), le bruit d'annulation (l'anti-bruit) et le résultat de superposition [2], [3].

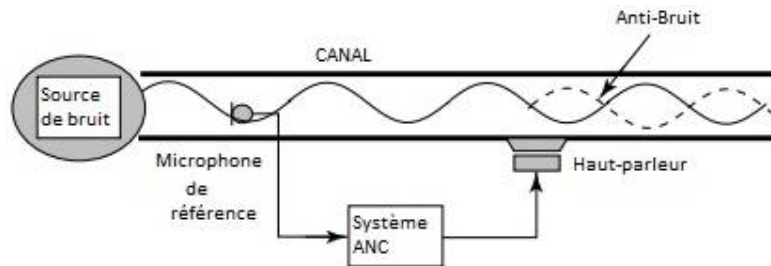


**Figure 1.1 :** Concept d'annulation de bruit actif [2].

Le contrôle actif du bruit s'accroît rapidement, car il permet d'améliorer significativement le contrôle du bruit, souvent avec des avantages potentiels en taille, volume et coût de calcul.

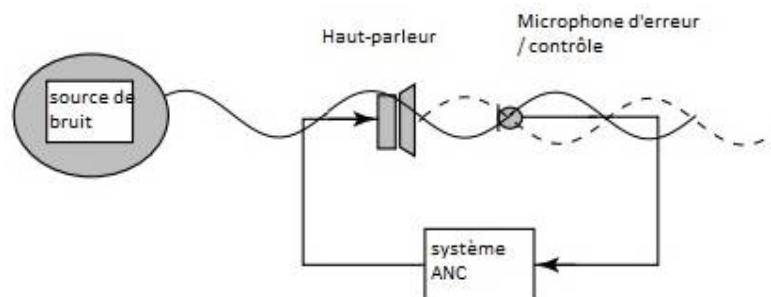
## 1.2 Concept du contrôle actif du bruit

La conception d'un système de contrôle actif du bruit (ANC : Active Noise Control) est basé sur l'utilisation d'un microphone et d'un haut-parleur à transmission électronique [2]. Dans les années trente, "Paul Lueg" a développé l'idée fondamentale de générer un son pour réduire un champ sonore indésirable et la notion de contrôle actif du bruit a été présentée dans cette époque sous forme d'un brevet. Lueg exposé le contrôle actif comme une propagation d'onde sonore dans un canal. Un microphone de référence détecte le son généré par une source de bruit. Le signal de sortie du microphone est filtré à travers une unité de commande avant d'être conduit vers le haut-parleur, ce qui génère le "anti-bruit" [4]. La figure 1.2 montre le concept du contrôle du bruit actif (ANC feed-forward).



**Figure 1.2 :** Concept du contrôle du bruit actif dans un canal présenté par P. Lueg [3].

Au début des années cinquante (1953) Harry Olson et Everett May introduisent et expérimentent l'ANC feedback dans le contrôle actif du bruit (figure 1.3).



**Figure 1.3 :** Contrôle actif du bruit basé sur le ANC feedback [3].

Contrairement au premier système, le microphone se situe après le haut-parleur qui détecte le champ sonore. Le signal de sortie du microphone est renvoyé à un contrôleur (système ANC) pour piloter le haut-parleur afin de produire une sortie qui entraîne une atténuation du champ sonore à proximité du microphone [3].

L'arrivée de techniques numériques a permis de produire des systèmes pratiques de contrôle actif du bruit basés sur le traitement du signal numérique et les dispositifs numériques. Kido et Chaplin ont effectué des travaux intensifs en explorant les applications des techniques de traitement du signal numérique et des dispositifs actifs de contrôle du bruit. Au début des années 80 (1980, 1981), Morgan, Burgess et Widrow travaillaient sur les algorithmes compensant le chemin physique entre les sources de contrôle et les capteurs (microphones). Ce chemin physique entraîne souvent une instabilité lors de l'utilisation de l'algorithme de type LMS (Least Mean Squares). En 1981, Widrow a présenté l'algorithme « Filtered-x LMS » (Filtered-x Least-Mean-Squares). Le concept de ce dernier était basé sur l'utilisation de sources de contrôle multiples et des capteurs pour contrôler le champ sonore dans un volume entier [2], [3].

### **1.3 Motivation**

Le contrôle de bruit acoustique utilise des techniques passives telles que les barrières et les silencieux pour atténuer le bruit indésirable. Ces techniques nécessitent des équipements qui sont relativement grandes, coûteuses, et inefficaces pour les basses fréquences. Par contre le système ANC atténue efficacement le bruit à basses fréquences contrairement aux méthodes passives. Le plus important est que le système ANC permet de faire un blocage sélectif avec des avantages potentiels en taille, volume et coût. Le blocage des basses fréquences est prioritaire puisque la plupart des bruits dans le monde réel sont inférieurs à 1 kHz. Cela nous a principalement conduits à se concentrer dans notre projet à l'annulation de bruit à basses fréquences.

# Chapitre 2 Notions acoustiques

---

## 2.1 Introduction

Le bruit acoustique est une nuisance sonore ennuyeuse qui affecte physiquement et physiologiquement les êtres humains. Avec le temps ce bruit peut causer des dommages irréversibles. Pour bien traiter et atténuer efficacement le bruit acoustique, il est important dans un premier temps de comprendre la propagation d'ondes acoustiques. Pour cela, ce chapitre va introduire brièvement quelques notions sur l'acoustique.

## 2.2 Acoustique

C'est une science qui étudie les phénomènes liés à la propagation des ondes sonores. Au 18<sup>ème</sup> siècle, **Joseph Sauveur** - physicien français- (1653-1716) inventa le terme "**acoustique**" pour désigner la **science de l'étude des sons**.

### 2.2.1 Son

Le son (ou le bruit) est le résultat de variations de pression, ou d'oscillations, dans un milieu élastique (air, eau, solides), généré par une surface vibrante ou un écoulement de fluide turbulent [5]. Plus généralement, il correspond à la propagation de perturbations mécaniques dans le milieu élastique. Ces perturbations sont perçues, entre autres, par l'oreille humaine qui les interprète comme des sons [6].

### 2.2.2 Ondes sonores

Quand les ondes sonores avancent, l'air se regroupe dans certains endroits et répartit dans d'autres régions, cela crée un motif alternatif de zones écrasées (compressions) et des zones étirées (diminutions). En d'autres termes, le son pousse et tire l'air en va-et-vient en même temps les ondes sonores font passer de l'énergie [6].

### 2.2.3 Vitesse de son

La vitesse de son (ou célérité) dépend des caractéristiques élastique et de la température absolue du milieu dans lequel l'onde acoustique se propage.

Dans l'air, la célérité se réduit à l'expression :

$$c = 20.05\sqrt{T} \text{ [m/s]} \quad (2.1)$$

Où  $T$  est la température en degrés kelvin ( $^{\circ}K$ ) et  $t$  est la température en degrés Celsius ( $^{\circ}C$ ).

Ci-dessous quelques valeurs de célérité sonores dans l'aire selon quelques températures :

Par exemple :

- à  $0^{\circ}C$  :  $c_0 = 330,45 \text{ m/s}$ .
- à  $20^{\circ}C$  :  $c_{20} = 342,35 \text{ m/s}$ .
- à  $100^{\circ}C$  :  $c_{100} = 386,25 \text{ m/s}$ .

Dans les métaux : Aluminium  $c = 5200 \text{ m/s}$  ; Acier  $c = 5100 \text{ m/s}$  ; Bronze  $c = 3000 \text{ m/s}$ .

Dans l'eau  $c_0 = 1410 \text{ m/s}$  [7].

### 2.2.4 Caractéristique de son

Le son est défini par de nombreux facteurs. Les plus importants sont les suivants :

- **La fréquence ( $f$ )** : Le nombre de cycles par seconde, et est exprimé en Hertz (Hz), Généralement le bruit composé de nombreuses fréquences combinées ensemble.
- **Longueur d'onde ( $\lambda$ )** : est la distance parcourue par l'onde de pression pendant un cycle.

- **Amplitude (A)** : L'amplitude est l'intensité maximale de la pression à n'importe quel point de l'onde sonore.
- **Vitesse de propagation** : on a déjà parlé (2.2.3 Vitesse de son ).
- **La période (T)** :C'est le temps pris pour qu'un cycle d'une onde passe un point fixe [5]. C'est lié à la fréquence par :  $T = 1 / f$ .

### 2.2.5 Vitesse vibratoire

Les molécules du milieu sont agitées d'une vitesse vibratoire fluctuante  $\vec{u}(t)$ , de valeur très petite par rapport à la célérité du milieu.

Cette vitesse  $\vec{u}(t)$  ayant une valeur moyenne nulle [8], on considère sa valeur efficace :

$$U_{eff} = \sqrt{\overline{u^2(t)}} \quad (2.2)$$

### 2.2.6 Pression sonore (acoustique)

Les sources sonores produisant en chaque point de l'espace une pression instantanée fluctuante, représentant les fluctuations de la source cette pression s'ajoute à la pression atmosphérique  $P_0$  (ambiante) mais de valeur très petite par rapport à elle. Pour une vibration harmonique [8], la pression sonore peut être calculée comme suit :

$$p(x, t) = P_{max} \sin(\omega t - kx) = P_T - P_0 \quad (2.3)$$

Où  $P_{max}$  est l'amplitude maximale,  $\omega$  est la pulsation en rad/s,  $t$  représente le temps en seconde,  $k$  désigne le vecteur d'onde,  $c$  est la célérité en mètre par seconde et  $\lambda$  représente la longueur d'onde.

La pression efficace  $p_{eff}$  entre les instant  $t_0$  et  $t_1$  est définie par :



$$p_{eff} = \sqrt{\frac{1}{t_1 - t_0} \int_{t_0}^{t_1} p^2(t) dt} \quad (2.4)$$

### 2.2.7 Intensité sonore

L'intensité sonore ou le flux d'énergie transporté par unité de surface, est la moyenne de produit de la pression et la vitesse ondulatoire, pris sur un temps plus long que les périodes de leurs fluctuations [7], elle exprimée en (W/m<sup>2</sup>) :

$$I = \overline{p \cdot u} \quad (2.5)$$

Si  $\emptyset$  est la différence entre  $p$  et  $u$  donc :

$$I = p_{eff} U_{eff} \cos(\emptyset) \quad (2.6)$$

### 2.2.8 Impédance acoustique

L'impédance acoustique est définie comme étant le rapport entre les valeurs instantanées de la pression et de la vitesse en un point ce rapport dépend du milieu de propagation et ses limites [7], [8], sa dimension est (kg/m<sup>2</sup>.s) :

$$Z = \frac{p}{u} = R + j \cdot X = |Z| \cdot e^{j\emptyset} \quad (2.7)$$

### 2.2.9 Niveau de pression sonore

En anglais SPL (Sound Pressure Level) est défini comme étant le logarithmique décimal du rapport entre la pression sonore et la pression de référence (seuil d'audibilité  $p_0 = 2 \cdot 10^{-5} Pa$ ) [7], [9] :

$$L_p = 20 \cdot \log\left(\frac{p_{eff}}{p_0}\right) \quad (2.8)$$

### 2.2.10 Niveau de puissance sonore

En anglais SWL (Sound Power Level) est défini comme étant le logarithmique décimal du rapport entre la puissance sonore et la puissance de référence (seuil d'audibilité  $W_0 = 10^{-12} Watt$ ) [9] :

$$L_0 = 10. \log \left( \frac{W}{W_0} \right) \quad (2.9)$$

Source	le niveau sonore en dB (A)
silence, commence à entendre le sens	0...10
Susurrement	20
silence arrière-cour	30
Discours amorti, silence à la maison	40
discours normal	50
discours fort, aspirateur	60
bruit de la rue	70
musique forte	80
bruit de rue fort	80...90
Klaxon, marteau	90...100
banc d'essai du moteur	100...110
banc d'essai du moteur	110...120
Hélice (distance de 5m)	120...130
Jet	130...150
Explosion	150...160

Tableau 2.1 : Valeurs typiques du niveau sonore [8].

Source	Puissance
voix humaine (discours normal)	Environ $7 \cdot 10^{-6} \text{ W}$
voix humaine (maximum)	Environ $2 \cdot 10^{-3} \text{ W}$
Violon	$10^{-3} \text{ W}$
grand piano	Environ $0.2 \text{ W}$
Trompette	environ $0.3 \text{ W}$
Organ	Jusqu'à $10 \text{ W}$
haut-parleur	Jusqu'à $100 \text{ W}$

**Tableau 2.2** : Puissance de différentes sources sonores [8].

# Chapitre 3 Contrôle actif du bruit

---

## 3.1 Introduction

En monde réel, on est habituellement exposé au bruit acoustique dans diverses conditions, tels que : l'industrie, les moyens de transport public, les hôpitaux, les avions, etc. Ces cas sont causés généralement par un grand nombre de sources (exp par moteurs, souffleurs, transformateurs, ventilateurs, compresseurs et systèmes de ventilation) [10].

Le bruit est simplement un son indésirable, l'exposition à des niveaux élevés de ce dernier peut même entraîner une perte auditive temporaire ou permanente.

## 3.2 Pourquoi le contrôle actif du bruit ?

Le contrôle actif du bruit (ANC) utilise le phénomène d'interférence des ondes : lorsque deux ondes avec la même amplitude et fréquence, mais inversées en phase se déplacent dans la même direction, elles se neutraliseront grâce à une interférence destructrice, ce qui conduit à l'annulation de bruit [8]. Les défis dans ce cas sont d'identifier le signal d'origine et de générer l'inverse sans retard dans toutes les directions où les bruits interagissent et superposent [11].

Fondamentalement, le système ANC possède une propriété acoustique et une propriété électrique. Ce système utilise un microphone et un haut-parleur à transmission électronique pour générer un son d'annulation (anti-bruit) [12].

Vu que les caractéristiques de la source de bruit acoustique et de l'environnement varient en fonction du temps (phénomène non stationnaire), le système ANC doit donc être adapté à ces variations. L'utilisation de filtres adaptés est le meilleur choix pour résoudre de tels problèmes. Un filtre adaptatif ajustera dans ce cas les coefficients pour minimiser un signal erreur. Ces filtres sont généralement réalisés par des coefficients

finis (réponse impulsionnelle finie "FIR"), et des coefficients infinis (réponse impulsionnelle infinie "IIR").

### 3.2.1 Composantes de système ANC

Un système ANC se compose des deux principales parties suivantes :

#### ➤ **Capteurs**

Les capteurs de vibration, les microphones ou d'autres dispositifs qui détectent la perturbation primaire et surveillent la performance du système de contrôle en mesurant l'erreur de validation.

Classification des microphones :

On peut classer les microphones selon plusieurs critères :

#### ➤ **D'après leur type**

Il existe plusieurs types de microphones mais on va citer que les plus connu :

- **Microphone électrodynamique**

Les microphones électromagnétiques sont constitués d'un conducteur bobiné sur un cylindre, ce dernier est collé à la membrane, qui le fait vibrer dans le fort champ magnétique fixe d'un aimant permanent. Le mouvement crée une force électromotrice créant le signal électrique (figure 5.3).



**Figure 3.1** : Microphone électrodynamique.

- **Microphone électrostatique**

La fidélité d'un microphone électrodynamique est liée à l'homogénéité du champ magnétique dans lequel se déplace la bobine, cette homogénéité est difficile à attendre, alors qu'un champ électrostatique peut être beaucoup plus homogène. Dans les microphones électrostatiques, la membrane, est l'une des armatures d'un condensateur, elle est couverte d'une mince couche conductrice, chargée par une tension continue, l'autre armature étant fixe. La vibration créera une variation dans la capacité en rapprochant et éloignant les armatures (figure 5.4).



**Figure 3.2 :** Microphone électrostatique.

- **Microphone électrostatique à électret**

Les microphones électrostatiques à électret conservent une charge électrostatique permanente, ils tirent parti d'une propriété de certains matériaux. D'un côté ce matériau constitue une armature de condensateur, et la membrane l'autre.



**Figure 3.3 :** Microphone électrostatique à électret.

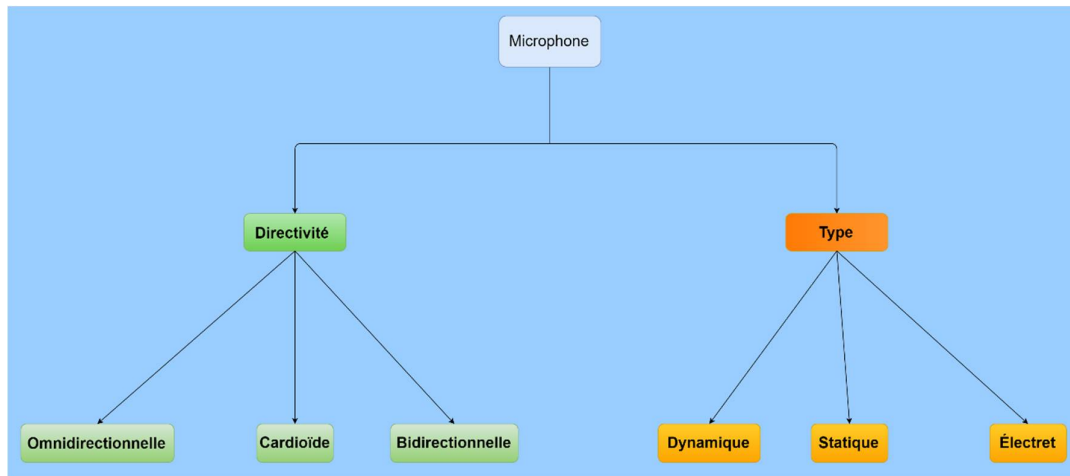
Type	Utilisations	Avantages	Inconvénients	Qualité	Prix
Dynamique	Toutes	Sensibilité élevée robustesse, Directivités variés, Courbe de réponse étendue		De médiocre à excellent	Peu couteux à très coûteux
Statique	Toutes	Haute-fidélité, nécessité de forte amplification	Nécessite une source de tension de polarisation	Excellent	Très coûteux
Electret	Toutes	Comme statique, mais sans polarisation	Craint parfois la chaleur et l'humidité	Excellent	Coûteux à très coûteux

**Tableau 3.1** : Comparaison entre les microphones selon leur type.

### ➤ D'après leur directivité

La directivité est une caractéristique très essentielle car elle représente la sensibilité du microphone en fonction de la provenance du son. On peut citer :

- **Omnidirectionnel**
- **Cardioïde**
- **Bidirectionnel**



**Figure 3.4** : Classification des microphones.

### ➤ **Actionneurs**

Les dispositifs qui font physiquement le travail de modification de la réponse impulsionnelle du système ANC, sont généralement des dispositifs électromécaniques tels que les haut-parleurs [8], [10].

### 3.2.2 Applications du système ANC

Aux basses fréquences, la technique du contrôle actif du bruit est un moyen efficace pour fournir une atténuation importante par rapport au contrôle du bruit passif .Il y'a un grand nombre d'applications où l'ANC est utilisé pour l'atténuation du bruit indésirable [13] :

#### ➤ **Appareils électroniques**

Un grand nombre d'appareils électroniques tels que les aspirateurs, les machines à laver et les conduits de climatisation utilisent la technique ANC pour atténuer les bruits de basse fréquence.

#### ➤ **Industrie**

Dans l'industrie, il existe une grande quantité de bruit des ventilateurs, des transformateurs, des compresseurs, etc. La technique de contrôle du bruit est largement



utilisée dans ce champ, soit des méthodes actives ou soit des méthodes passives ou leurs combinaisons.

➤ ***Automobiles et transport***

De nos jours, le problème de la réduction du bruit dans l'automobile et dans le transport est pris au sérieux. Beaucoup de travaux sont effectués pour fournir une atténuation du bruit dans les voitures, les camions, les trains, avions, les hélicoptères et surtout dans les engins militaires tels que les tanks [1].

### **3.3 Systèmes du contrôle actif du bruit**

L'annulation du bruit à large bande requiert la connaissance de la source de bruit (le bruit primaire) afin de générer le signal anti-bruit. La mesure du bruit primaire est utilisée comme entrée de référence pour l'annulateur du bruit. Le bruit primaire qui est en corrélation avec le signal d'entrée de référence est annulé en aval du générateur du bruit (un haut-parleur) lorsque la phase et amplitude sont correctement modélisées dans le contrôleur numérique.

Pour l'annulation du bruit à bande étroite (réduction des bruits périodiques causés par les machines en rotation), des techniques actives ont été développées très efficaces et ne dépendent pas de la causalité (ayant une connaissance préalable du signal de bruit). Au lieu d'utiliser un microphone d'entrée, un signal de tachymètre fournit des informations sur la fréquence primaire du générateur de bruit. Parce que tout le bruit répétitif se produit à l'harmonique de la fréquence de rotation basique de la machine, le système de contrôle peut modéliser ces fréquences de bruit connues et générer le signal anti-bruit. Ce type de système de commande est souhaitable dans une cabine de véhicule, car cela n'affectera pas les signaux d'avertissement du véhicule, les performances radio ou la parole, qui ne sont normalement pas synchronisés avec la rotation du moteur.

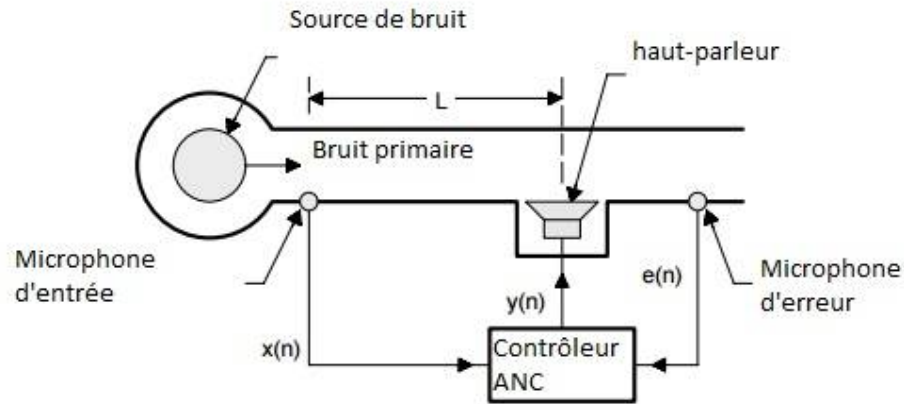
Les systèmes actifs de contrôle du bruit sont basés sur l'une des deux méthodes. Le contrôle Feed-forward est l'endroit où une entrée de bruit de référence cohérente est détectée avant de se propager après le haut-parleur d'annulation. Le contrôle de feed-

back [14], [15] est l'endroit où le contrôleur de bruit actif tente d'annuler le bruit sans bénéficier d'une entrée de référence en amont.

### **3.3.1 Feed-forward contrôle actif du bruit**

#### ***a. à large bande***

Une quantité importante de bruit à large bande est produite dans les systèmes réels tels que les chauffages et les climatisations. Un système ANC simple pour un canal à large bande est illustré par la figure 3.1. Il dispose d'un seul capteur de référence, d'une seule source secondaire et d'un seul capteur d'erreur. Un microphone de référence est placé près de la source du bruit pour détecter le bruit produit par celui-ci et génère la durée de la version continue du signal de référence  $x(n)$  avant qu'il n'atteigne le haut-parleur (source secondaire). Le système ANC traite le signal de référence afin de produire un signal de commande pour diriger le haut-parleur. Un micro d'erreur est placé en aval du haut-parleur. Ce microphone détecte le bruit produit par la concaténation des deux signaux pour que le ANC puisse mettre à jour les coefficients de ce filtre afin de produire un signal plus identique au signal de référence. L'objectif principal du contrôleur ANC est de minimiser le bruit dans le conduit en générant le soi-disant anti-son  $y(n)$  via le haut-parleur qui est en mode 180 degrés mais égale en grandeur avec le son primaire lorsqu'il atteint le microphone d'erreur. Le contrôleur ANC s'ajuste automatiquement en fonction des informations fournies par les signaux de référence et d'erreur pour minimiser le bruit acoustique détecté par le microphone d'erreur dans le sens du minimum du moindre carré [1].



**Figure 3.5** : Système ANC Feed-forward à large bande dans un canal [2].

***b. à bande étroite***

Les bruits de ce type sont généralement générés par les moteurs, les compresseurs, les hélices et les ventilateurs qui sont en principe du type périodique. En utilisant des capteurs appropriés, nous pouvons observer directement le mouvement mécanique de ces sources qui produisent un tel bruit. Les capteurs utilisés pour observer le mouvement nous donnent l'information sous la forme de signaux électriques qui contiennent toutes les harmoniques du bruit généré primaire et de la fréquence fondamentale. Le diagramme de blocs est illustré sur la figure 3.2, qui explique l'ANC à bande étroite pour atténuer le bruit dans un conduit. En ANC à bande étroite, le microphone de référence peut être remplacé favorablement par un capteur non acoustique qui élimine les éventuels problèmes de rétroaction acoustique. Un micro d'erreur est toujours nécessaire pour mesurer le bruit à contrôler pour ajuster les coefficients du filtre adaptatif afin de minimiser l'erreur de la moyenne-carrée [2].

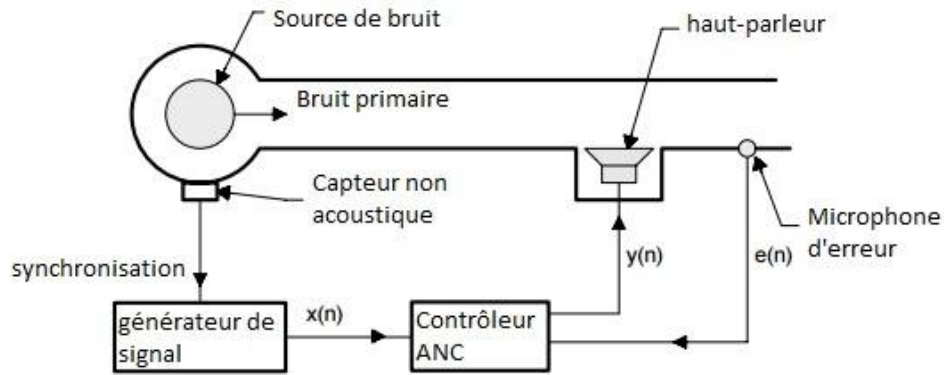


Figure 3.6 : Système ANC Feed-forward à bande étroite [2].

### 3.3.2 Feed-back contrôle actif du bruit

Le contrôle actif du bruit de feed-back a été proposé par Olson et May en 1953 [14]. Le principe est schématisé sur la figure 3.3. Dans ce schéma, un microphone est utilisé comme capteur d'erreur pour détecter le bruit indésirable. Le signal du capteur d'erreur est renvoyé par un amplificateur (filtre électronique) avec une amplitude et une réponse de phase conçues pour produire une annulation au capteur via un haut-parleur situé près du microphone. Cette configuration ne fournit qu'une atténuation limitée sur une plage de fréquences restreinte pour des sons périodiques ou à limitation de bande. Il souffre également de l'instabilité. Cependant, en raison de la nature prévisible des signaux à bande étroite, un système plus robuste qui utilise la sortie du capteur d'erreur pour prédire l'entrée de référence a été développé (voir la figure 3.3). L'entrée de référence régénérée est combinée avec le système de contrôle du bruit actif à bande étroite.

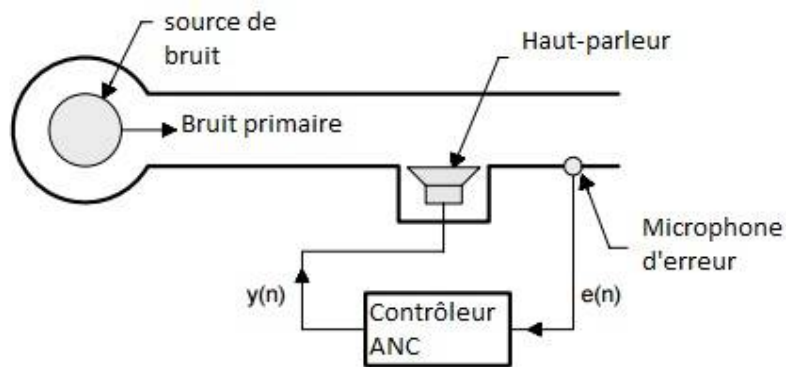


Figure 3.7 : Système ANC Feed-back [14].

# Chapitre 4 Algorithmes adaptatifs pour le contrôle actif du bruit

---

## 4.1 Introduction

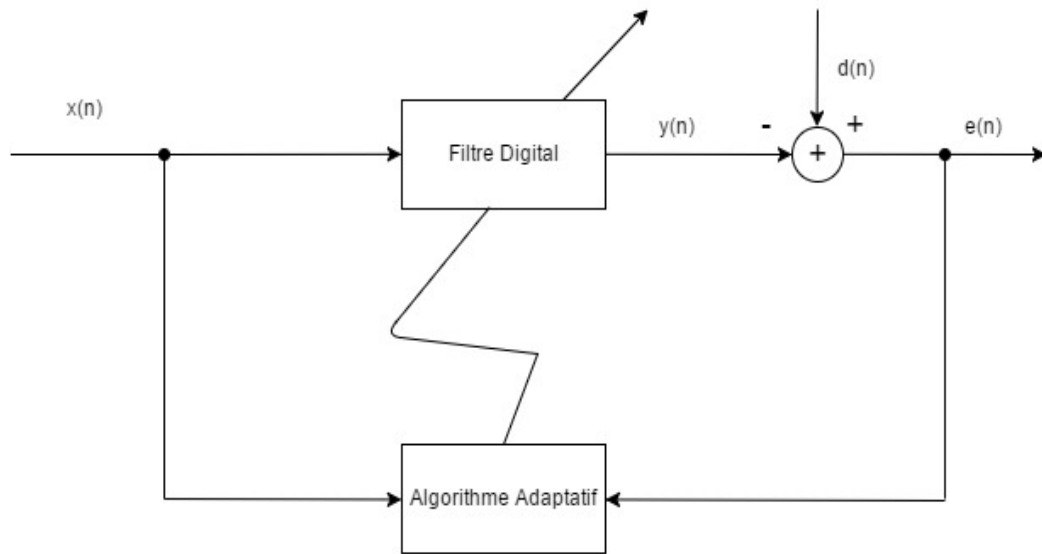
Le système ANC se compose d'un filtre adaptatif qui comporte deux éléments essentiels. Ces deux parties comprennent un filtre numérique réglable et un algorithme de contrôle. Le filtre numérique effectue le traitement du signal requis tandis que l'algorithme de contrôle ajuste les coefficients du filtre numérique. Les algorithmes de filtrage adaptatif jouent un rôle important dans les systèmes ANC car ils sont généralement utilisés dans un système tel qu'un contrôleur adaptatif. Compte tenu de l'importance des algorithmes de contrôle, ce chapitre met en évidence les algorithmes de contrôle les plus simple qui ont été déployés en tant que contrôleurs adaptatifs dans les systèmes ANC [1].

## 4.2 Algorithmes adaptatifs

Les filtres adaptatifs sont des systèmes appliqués sur des données bruitées, ils sont capable de changer de comportement en fonction de contexte d'exécution afin d'obtenir une information utile a un certain temps discret  $n$ . La figure 4.1 représente le schéma bloc d'un filtre adaptatif. Dans cette figure  $d(n)$  représente le signal désiré,  $y(n)$  est le signal de sortie et  $e(n)$  est le signal erreur.

L'erreur est définit mathématiquement par :

$$e(n) = d(n) - y(n) \quad (4.1)$$



**Figure 4.1** : Structure de filtrage adaptatif.

On peut classer les filtres adaptatifs selon leur réponse impulsionnelle :

- RIF (filtre à réponse impulsionnelle finie) : ce type de filtre est le plus utilisé à cause de l'absence de pôles qui lui rend inconditionnellement stable.
- RII (filtre à réponse impulsionnelle infinie) : ce type de filtre est le plus proche de la réalité physique, il est rarement utilisé à cause des risques d'instabilité et de complexité.

Concernant le choix de l'algorithme adaptatif se reposer sur plusieurs critères, les plus importants sont :

- rapidité de convergence,
- stabilité,
- robustesse au bruit,
- complexité.

## 4.2.1 Algorithme LMS

Il a été développé par « B. Widrow, M.E. Hoff » en 1959, et depuis il est devenu l'algorithme le plus utilisé à cause de ces performances en termes de simplicité et complexité. Il est aussi connu comme l'algorithme de gradient stochastique, il se base sur l'estimation de la fonction d'autocorrélation et d'inter-corrélation (figure 4.2).

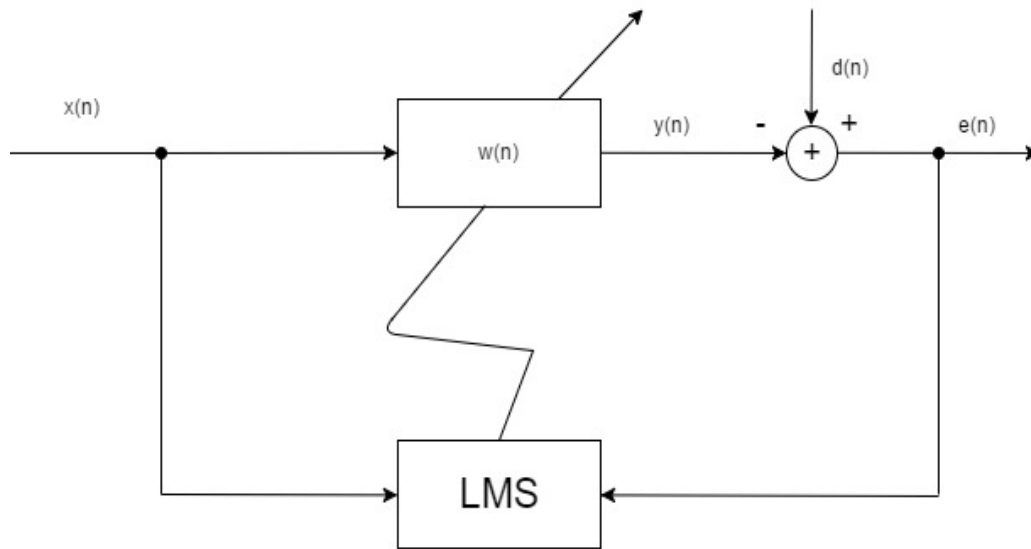


Figure 4.2 : Bloc diagramme de l'algorithme LMS.

Cet algorithme nécessite trois étapes essentielles pour chaque itération :

- Calcul de la sortie :

$$y(n) = \sum_{i=0}^{N-1} w_i(n)x(n-i) = \mathbf{w}^T(n) \mathbf{x}(n) \quad (4.3)$$

- Estimation de l'erreur :

$$e(n) = d(n) - y(n) \quad (4.4)$$

- Mise à jour des coefficients du filtre :

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n) \mathbf{x}(n) \quad (4.5)$$

Où  $\mathbf{x}(n)$  représente le vecteur du signal d'entrée  $\mathbf{x}(n) = [x(0)x(1)x(2) \dots x(n - N + 1)]^T$ ,  $\mathbf{w}(n) = [w_0(n)w_1(n)w_2(n) \dots w_{N-1}(n)]^T$  sont les coefficients du filtre adaptatif à l'instant  $n$ , et  $\mu$  est le pas d'adaptation compris entre  $]0 \frac{1}{\lambda_{max}}[$  pour satisfaire les conditions de convergence [16] (où  $\lambda_{max}$  est la plus grande valeur propre de la matrice d'auto-corrélation de l'entrée  $x(n)$ ).

### 4.2.2 Algorithme LMS normalisé (NLMS)

La convergence de l'algorithme LMS est relative à au pas d'adaptation  $\mu$ , qui est lui-même lié aux données de l'entrée  $x(n)$  ( $\lambda_{max}$ ), par conséquent ces performances varient avec l'entrée  $x(n)$ .

Pour détourner cette inconvénient, l'algorithme LMS est normalisé par l'énergie du signal d'entrée. D'où l'algorithme NLMS nécessitera une étape supplémentaire, en augmentant sa complexité de  $N$  multiplications, mais acceptable considérant les performances acquis par cette modification [16].

Les étapes de l'algorithme NLMS sont :

- Calcule de la sortie :

$$y(n) = \sum_{i=0}^{N-1} w_i(n)x(n-i) = \mathbf{w}^T(n) \mathbf{x}(n) \quad (4.6)$$

- Estimation de l'erreur :

$$e(n) = d(n) - y(n) \quad (4.7)$$

- Calcule du pas d'adaptation :

$$\mu(n) = \frac{\mu}{\mathbf{x}^T(n)\mathbf{x}(n)} \quad (4.8)$$

- Mise à jour des coefficients du filtre :

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\mathbf{x}(n) \quad (4.9)$$



### 4.2.3 Algorithme Leaky LMS (LLMS)

Conçu pour surmonter la lente convergence de l'algorithme LMS en cas de propagation de valeurs propres élevées. L'algorithme utilise l'heuristique de punition/récompense avec une fonction d'ajustement de fuite quantifiée pour faire varier la fuite. Les résultats des recherches confirment que le nouvel algorithme peut surpasser de manière significative le LMS standard lorsque la propagation de la valeur propre de l'entrée est élevée. Il a été démontré qu'il peut être utilisé pour améliorer la stabilité dans une implémentation de précision finie, améliorer les effets de l'excitation non persistante et réduire les effets indésirables comme étalement de spectre [17].

La différence entre les deux algorithmes est dans la mise à jour des coefficients du filtre. Au leaky LMS on ajoute le paramètre  $\gamma$  appelé «fuite» :

$$\mathbf{w}(n + 1) = (1 - \mu\gamma) \mathbf{w}(n) + \mu e(n)\mathbf{x}(n) \quad (\gamma \geq 0) \quad (4.10)$$

### 4.2.4 Algorithme Filtré-x LMS

L'utilisation du filtre adaptatif pour un système ANC, est très difficile à cause de la présence du chemin secondaire comme le montre les deux figures (figure 4.3 et figure 4.4)

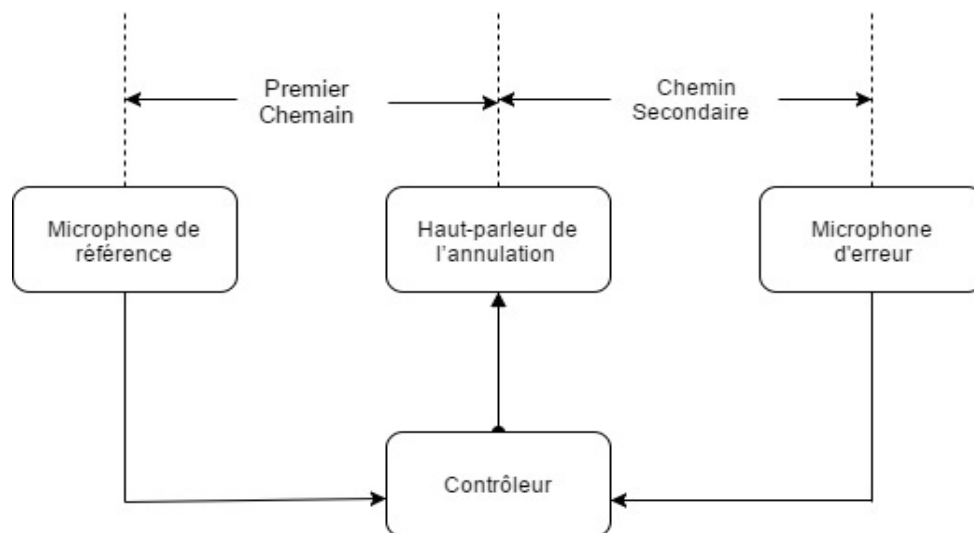


Figure 4.3 : Modélisation des chemins dans un système ANC.

Le bruit primaire  $d(n)$  est combiné avec la sortie anti-bruit  $y(n)$  du filtre adaptatif. Le signal anti-bruit peut être modifié par la fonction du chemin secondaire  $s(z)$  dans le canal acoustique de  $y(n)$  à  $e(n)$  (figure 4.4), et de même que le bruit principal est modifié par le chemin primaire  $p(z)$ . Par conséquent,

$$e(z) = d(z) + y(z)s(z) \quad (4.11)$$

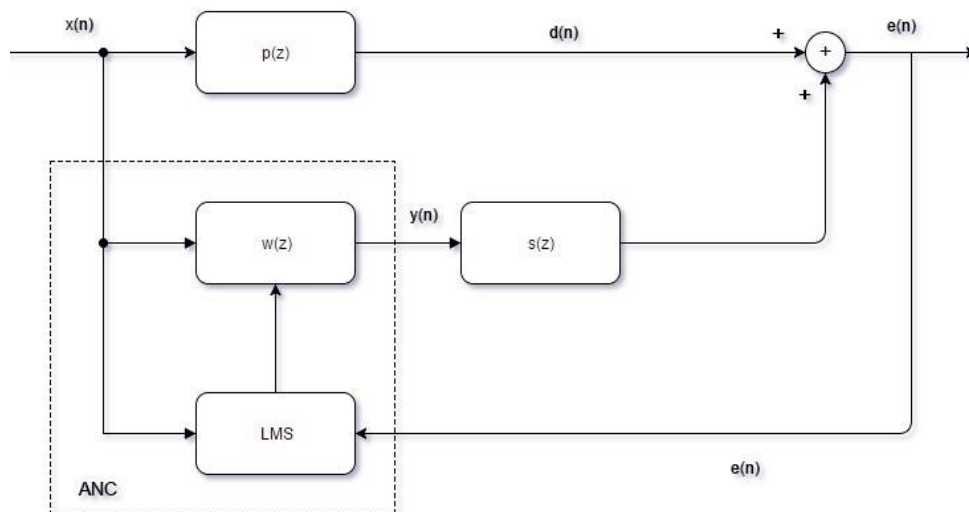
$$e(z) = x(z)p(z) + x(z)w(z)s(z) \quad (4.12)$$

En supposant que  $w(z)$  a un ordre suffisant après la convergence du filtre adaptatif, l'erreur résiduelle sera nulle. Ce résultat nécessite que  $w(z)$  soit :

$$w(z) = -\frac{p(z)}{s(z)}. \quad (4.13)$$

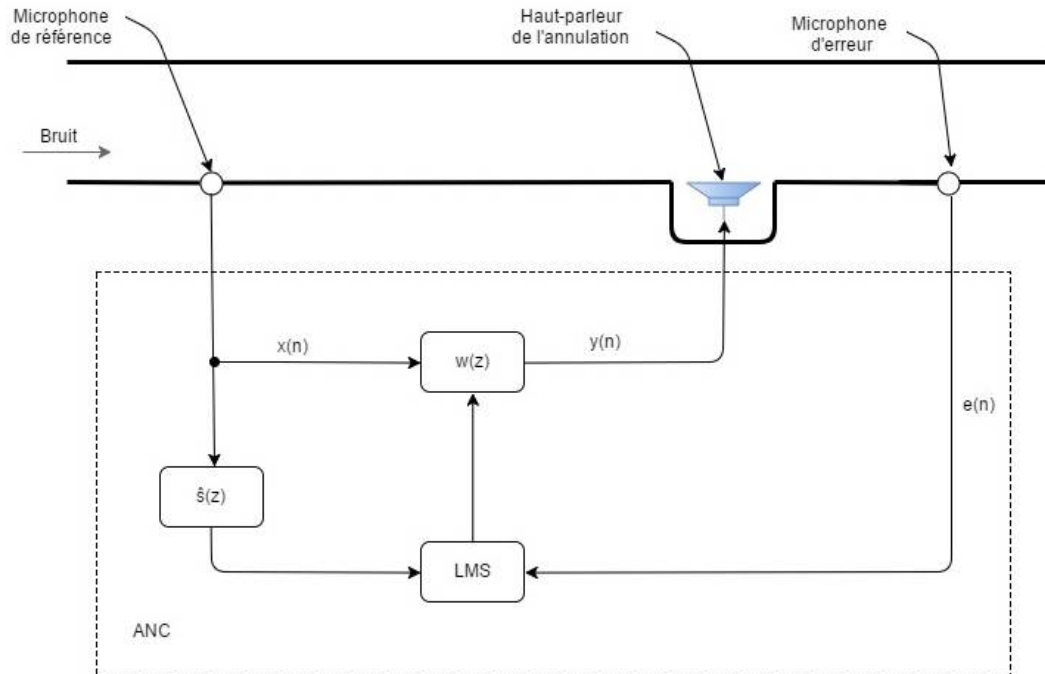
Ainsi, le filtre adaptatif  $p(z)$  doit modéliser le chemin primaire  $w(z)$  et inversement modéliser le chemin secondaire  $s(z)$ .

En outre, le système de contrôle est instable s'il existe une fréquence  $\omega$  telle que  $s(\omega) = 0$ , et il est inefficace s'il y a une fréquence  $\omega$  où  $p(\omega) = 0$ . Par conséquent, les caractéristiques du chemin secondaire  $s(z)$  auront des effets significatifs sur les performances d'un système ANC [2].



**Figure 4.4 :** Bloc diagramme de l'algorithme LMS dans l'ANC.

Pour assurer la convergence de l'algorithme, l'entrée est filtrée par un filtre d'estimation du chemin secondaire  $\hat{s}(z)$ . Il en résulte le filtre X-LMS (FX-LMS), pour compenser les effets du chemin secondaire dans les applications ANC.

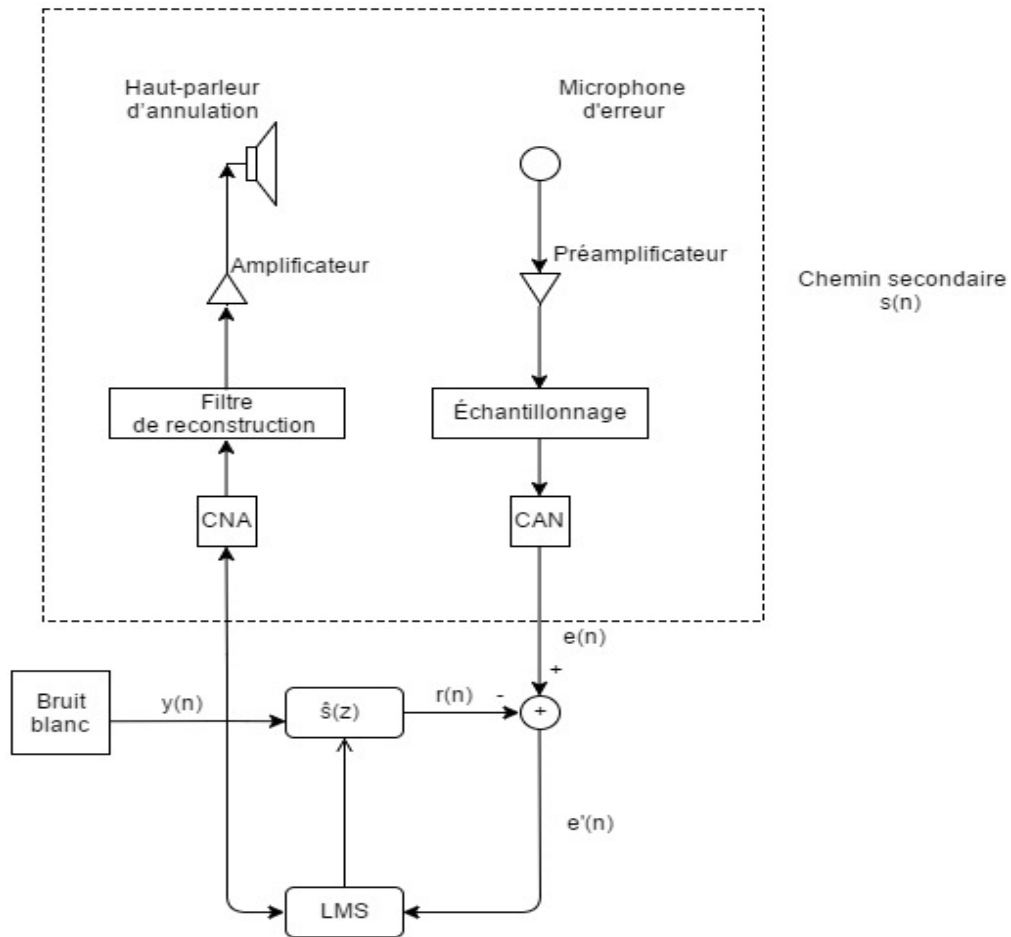


**Figure 4.5 :** Système ANC implémente avec l'algorithme FX-LMS.

Principalement l'algorithme FX-LMS est décomposé en deux étapes [18]:

**a. Configuration hors-ligne du filtre  $\hat{s}(z)$**

La configuration expérimentale pour la modélisation du système hors-ligne est illustrée par la figure 4.6, où un bruit blanc non corrélé est généré en interne.



**Figure 4.6 :** Modélisation de la configuration hors-ligne.

La procédure de formation est résumée par les étapes suivantes :

- Générer un échantillon de bruit blanc  $y(n)$ . Ce bruit est utilisé comme référence d'entrée pour le filtre adaptatif  $\hat{s}(z)$  et l'algorithme LMS, et il est conduit vers le haut-parleur de l'annulation.
- Calculer la réponse du filtre  $\hat{s}(z)$  :

$$r(n) = \sum_{i=0}^M \hat{s}_i(n) y(n-i) \quad (4.14)$$

Où  $\hat{s}_i(n)$  sont les coefficients du filtre  $\hat{s}(z)$  à l'instant  $n$  et  $M$  est l'ordre du filtre.

- Calculer l'erreur  $e'(n)$  :

$$e'(n) = e(n) - r(n). \quad (4.15)$$

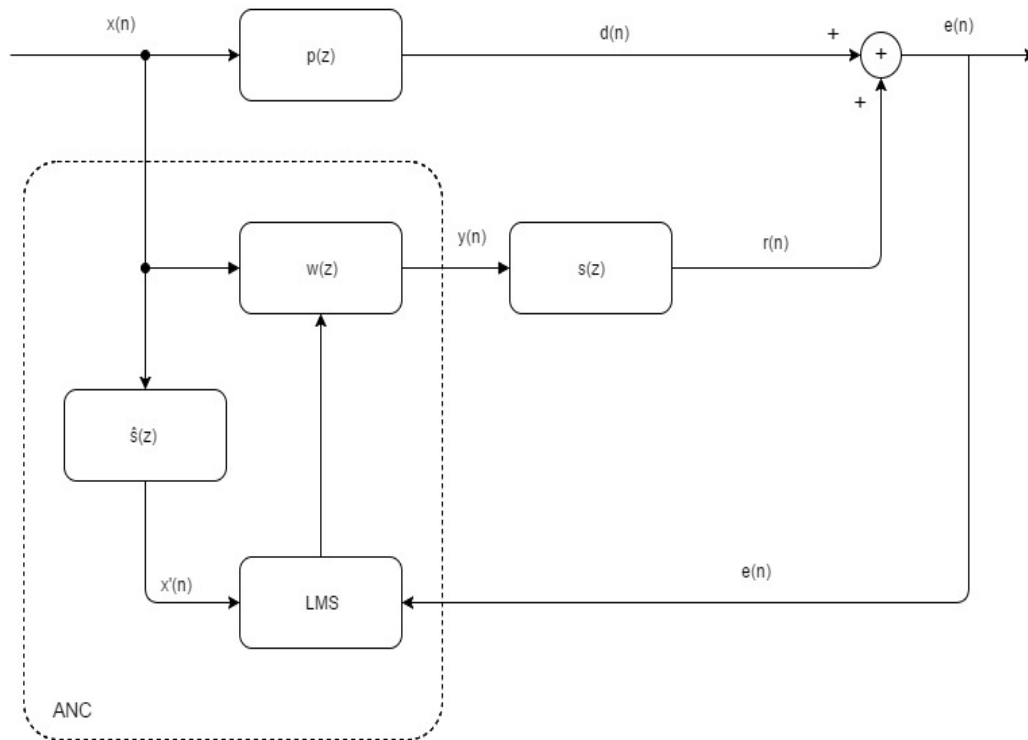
- Mettre à jour les coefficients du filtre  $\hat{s}(z)$  :

$$\hat{s}_i(n + 1) = \hat{s}_i(n) + \mu e'(n)y(n) \quad (4.16)$$

Avec :  $0 < \mu < \frac{1}{N P_y}$ ,  $P_y$  : l'énergie du signal  $y(n)$  et  $N$  est l'ordre du filtre  $\hat{s}(z)$ .

### b. L'annulation du bruit

La figure 4.7 représente le schéma bloc de l'algorithme FX-LMS avec le système ANC.



**Figure 4.7** : Bloc diagramme de l'algorithme FX-LMS avec le système ANC.

Dans ce cas, les étapes à suivre sont :

- Lire le bruit  $x(n)$  à partir du microphone de référence.
- Sélectionner l'ordre du filtre  $w(z)$  et trouver ces coefficients.

➤ Calculer la sortie  $y(n)$  :

$$y(n) = \sum_{i=0}^{N-1} w_i(n) x(n-i) = \mathbf{w}^T(n)\mathbf{x}(n) \quad (4.17)$$

➤ Calcule du  $x'(n)$  :

$$x'(n) = \sum_{i=0}^M \hat{s}_i(n) x(n-i) \quad (4.18)$$

➤ Mettre à jour les coefficients du filtre  $w(z)$  :

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu e(n)\mathbf{x}'(n) \quad (4.19)$$

➤ Répétez à partir de la troisième étape jusqu'à ce que le bruit soit annulé.

#### 4.2.5 Filtré-X-NLMS

En s'inspirant des étapes de l'algorithme FX-LMS, on obtient l'algorithme FX-NLMS suivant :

➤ L'équation de la mise à jour de la configuration hors-ligne :

$$\hat{s}_i(n+1) = \hat{s}_i(n) + \frac{\mu}{\mathbf{y}^T(n)\mathbf{y}(n)} e'(n)\mathbf{y}(n) \quad (4.20)$$

➤ L'équation de la mise à jour de l'annulation du bruit :

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{\mu}{\mathbf{x}'^T(n)\mathbf{x}'(n)} e(n)\mathbf{x}'(n) \quad (4.21)$$

#### 4.2.6 Algorithme Leaky FX-LMS

Dans l'algorithme Leaky FX-LMS, il n'aura de modification que juste dans la fonction de la mise à jour du filtre :

$$\mathbf{w}(n+1) = (1 - \mu\gamma)\mathbf{w}(n) - \mu e(n)\mathbf{x}'(n) \quad \gamma \geq 0 \quad (4.22)$$

### 4.3 Choix de l'algorithme

Dans cette section on va comparer les algorithmes décrits précédemment. La comparaison sera réalisée sous forme de simulations. Pour le chemin secondaire et primaire on a généré un filtre d'ordre 14.

L'ordre choisi pour les filtres  $w(z)$  et  $\hat{s}(z)$  est fixé à 16. Il est important de souligner que nous n'avons pas testé l'effet de l'ordre du filtre sur les algorithmes car le nombre des coefficients du chemin secondaire est connu et fixé. Cependant, dans la pratique on a varié l'ordre jusqu'à obtention d'une bonne convergence de l'erreur vers l'optimal.

Pour la phase du test, deux étapes sont à suivre :

- Détermination du pas d'adaptation

Le critère du choix du pas d'adaptation est la convergence de l'erreur vers zéro. Ce pas d'adaptation est donné par la formule suivante :

$$0 < \mu < \frac{2}{N \cdot E[x(n)]}$$

$N$  : l'ordre du filtre.

$E[x(n)]$  : est l'énergie du signal d'entrée  $x(n)$ .

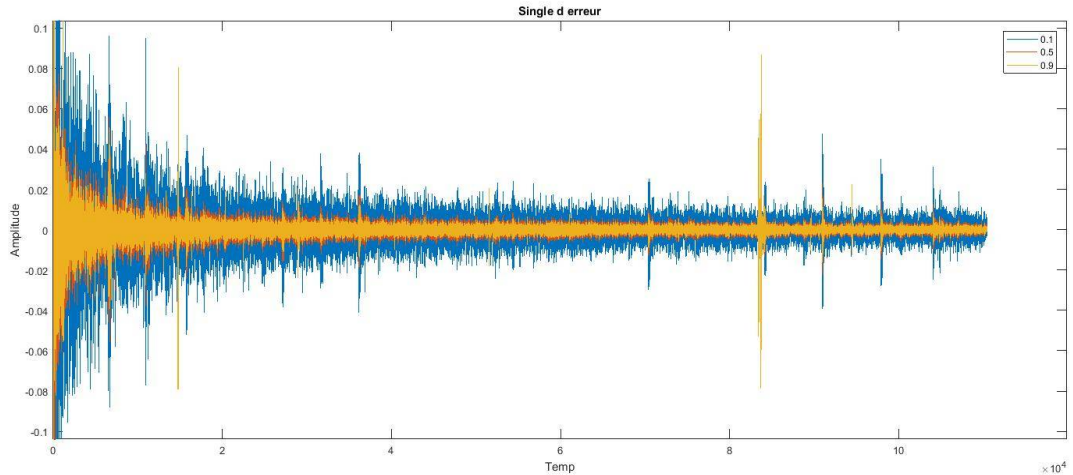
- Choix du bruit

On a pris un bruit d'un ventilateur d'une fréquence d'échantillonnage égale à 22050 Hz.

### 4.4 Tests des algorithmes

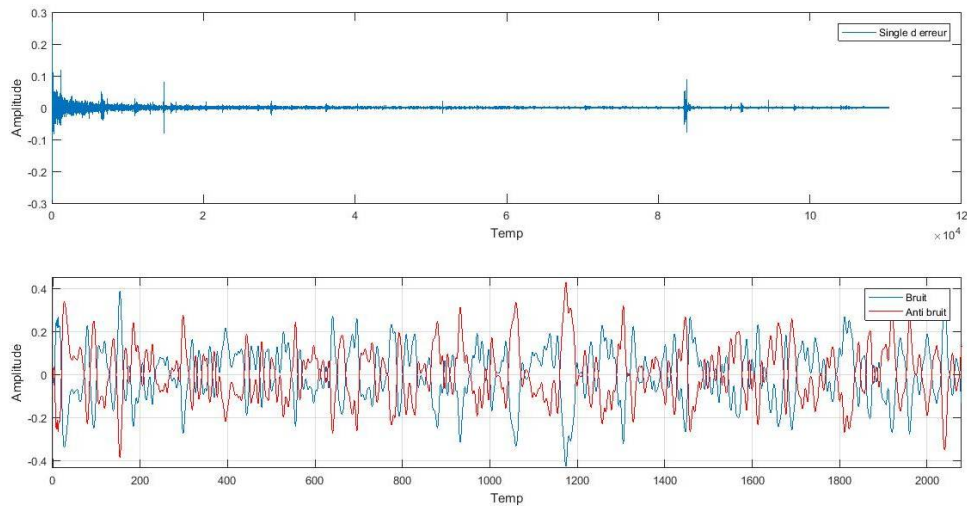
#### 4.4.1 FX-LMS

On peut constater d'après la figure 4.8 que le pas d'adaptation permet d'accélérer la convergence de l'algorithme FX-LMS.



**Figure 4.8** : Convergence de l'erreur de filtrage de l'algorithme FX-LMS pour différents pas d'adaptation.

Pour le test du, un bruit de ventilateur a été filtré par le chemin primaire pour obtenir le signal bruité. L'énergie maximale de ce dernier est estimée à  $-17.7573$  dB. Après l'application de l'algorithme on a obtenu un signal anti-bruit qui a la même amplitude que le signal bruité mais avec une phase opposée (voir figure 4.9). L'énergie du signal résultant est de  $-60.3504$  dB, d'où une diminution de plus de 40 dB.



**Figure 4.9** : Simulation de FX-LMS avec le bruit d'un ventilateur.



Le temps de convergence de l'algorithme est estimé à 2.57 seconds (le temps de calcul de l'estimation de chemin secondaire est inclut).

La figure 4.10 confirme la robustesse de l'algorithme montre lors de l'application du ANC basé sur FX-LMS.

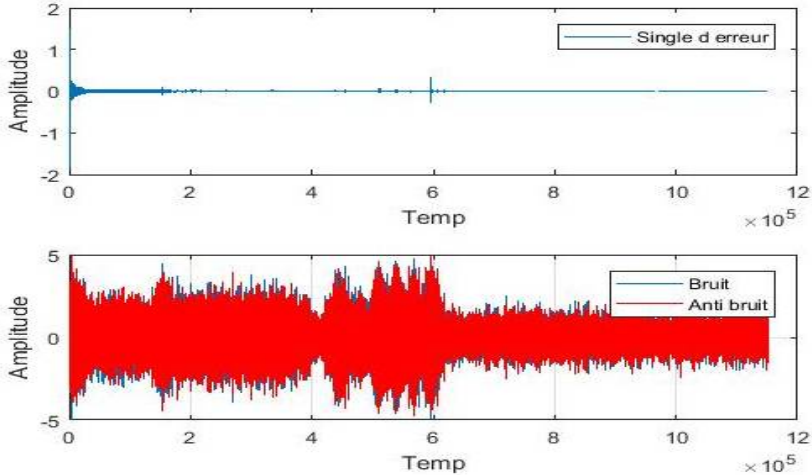


Figure 4.10 : Simulation de FX-LMS avec le bruit mixé,  $\mu=0.01$ .

### 4.4.2 FX-NLMS

La figure 4.11 montre l'effet du pas d'adaptation sur l'algorithme FX-NLMS.

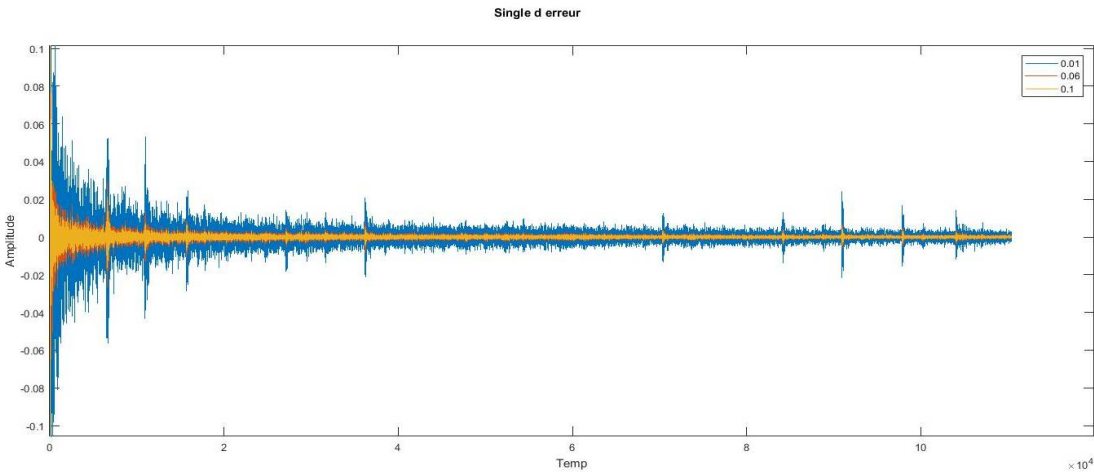
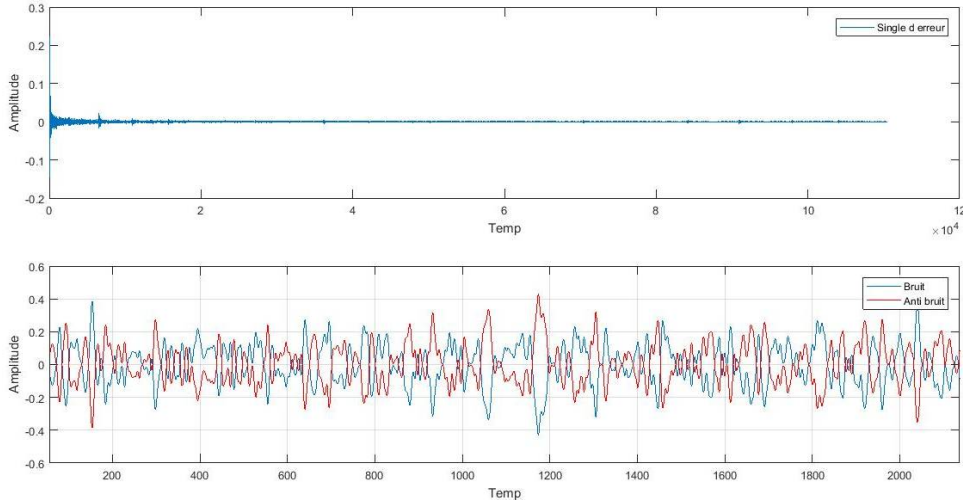


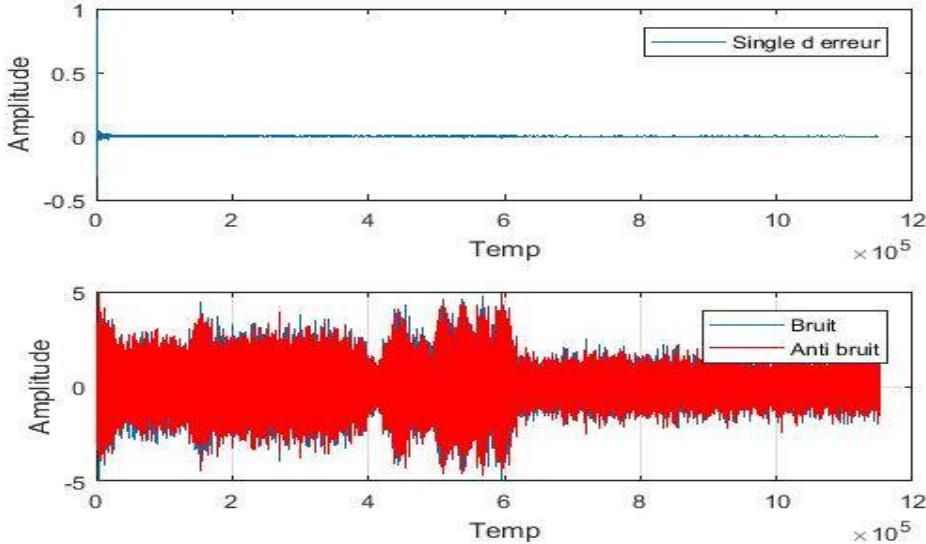
Figure 4.11 : Convergence de l'erreur de filtrage de l'algorithme FX-NLMS pour différents pas d'adaptation.

Selon la figure 4.12 on remarque que l'algorithme FX-NLMS converge rapidement. Aussi, le FX-NLMS permet d'atténuer le bruit jusqu'à plus de 52dB.



**Figure 4.12** : Simulation de FXNLMS avec le bruit d'un ventilateur.

D'après la figure 4.13, robustesse l'algorithme FX-NLMS est assuré pour une meilleure atténuation (53 dB).



**Figure 4.13** : Simulation de FX-NLMS avec le bruit mixé,  $\mu=0.19$ .

## 4.5 Conclusion

Dans ce chapitre, nous avons introduit trois types d'algorithmes adaptatifs pour le contrôle actif du bruit, à savoir : LMS, NLMS et Leaky LMS. Les versions filtrées de ces algorithmes ont été aussi introduites : FX-LMS, FX-NLMS et FX-Leaky LMS. D'après la littérature, on a constaté que les algorithmes FX-LMS, FX-NLMS sont les plus robustes et les plus utilisés pour un système ANC. Pour cela, on a opté pour des comparaisons sur ces derniers. Le résultat de test a favorisé l'algorithme FX-NLMS par rapport à l'algorithme FX-LMS, vue les bons résultats obtenus.

# Chapitre 5 Implémentation sur Raspberry PI

---

## 5.1 Introduction

Dans ce chapitre nous allons décrire les étapes suivies dans l'implémentation du système ANC sur la carte Raspberry PI. Pour l'architecture de l'ANC, nous utiliserons le feedback pour éviter le problème de synchronisation avec le PC. L'algorithme choisi pour cette partie est le FX-LMS et cela est dû à sa simplicité d'implémentation.

## 5.2 Matériel utilisé

### 5.2.1 Software

Dans notre projet nous avons choisi d'utiliser le SIMULINK de MATLAB comme un outil de conception. Nous avons utilisé la version MATLAB 2017a car à partir de la version MATLAB 2016b, MATHWORKS a intégré l'ensemble de support SIMULINK pour la carte Raspberry PI 3 model B.

### 5.2.2 Hardware

#### *a. Présentation de Raspberry PI 3*

Développée au Royaume-Uni par la Raspberry PI fondation (lancé en 2012). Il s'agit d'un nano-ordinateur mono-carte qui est aussi gros qu'une carte de crédit. Son but est de faciliter l'apprentissage de la programmation informatique [19]. Mais ce concept a volé l'attention de beaucoup de chercheurs dans le monde pour son design et son utilisation distinguée.

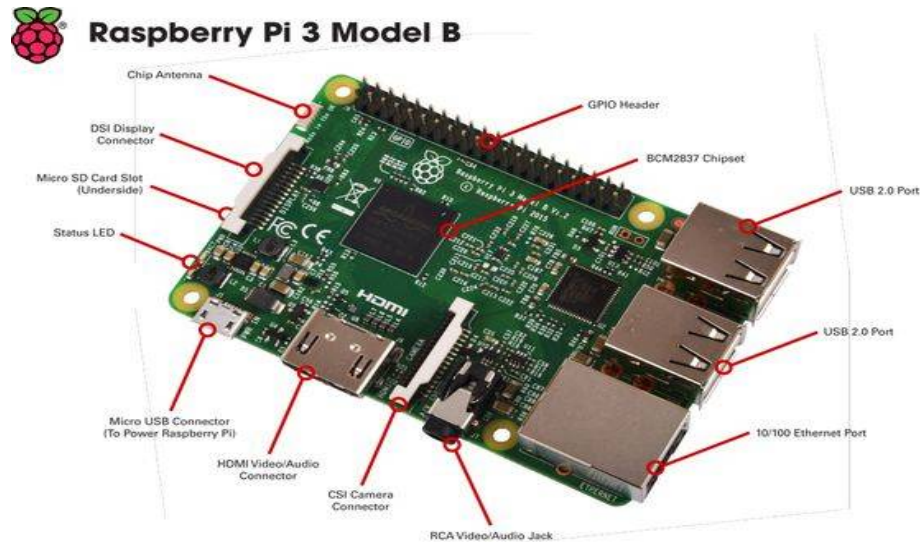






















Figure 5.1 : Raspberry PI 3.

**b. Compositions et caractéristique technique**

Pour le model 3B, il est une carte mère et il contient [19] :

	<b>Raspberry PI 3 Model B</b>
SoC	Broadcom BCM2837
GPU	Broadcom VideoCore IV, OpenGL ES 2.0, MPEG-2 et VC-1 (avec licence), 1080p30 h.264/MPEG-4 AVC high-profile décodeur et encodeur
CPU	1,2 GHzquadricœur ARM Cortex-A53
Mémoire	1 Go
Nombre de ports USB 2.0	4
Sorties Vidéos	HDMI et Composite (via un connecteur Jack)
Sorties audio	stéréo Jack 3,5 mm (sortie son 5.1 sur la prise HDMI) et Composite
Unité de lecteur/écriture	MicroSD62
Carte/connecteur réseau	10/100 Ethernet, Wifi 802.11n, Bluetooth 4.1
Périphériques bas niveau	17 × GPIO, UART, I <sup>2</sup> C bus, SPI bus avec deux chip selects, I <sup>2</sup> Saudio, +3.3 V, +5 V
Puissance nominal	800 mA (4 W)
Consommation maximale mesurée	720 mA
Source d'alimentation	5 volts via Micro-B USB ou GPIO header
Dimension	85,60 mm × 53,98 mm × 17 mm
Poids	45 g

Tableau 5.1 : Caractéristiques techniques.

Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I <sup>2</sup> C)		DC Power 5v	04
05	GPIO03 (SCL1 , I <sup>2</sup> C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I <sup>2</sup> C ID EEPROM)		(I <sup>2</sup> C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

**Figure 5.2** : Description des en-têtes GPIO.

### ***c. Avantages du Raspberry PI***

La carte Raspberry PI se distingue par la facilité du développement dans sa plateforme, car premièrement on peut installer plusieurs OS différents (Snappy Ubuntu core, SolydX RPI, Windows 10 IoT, Debian GNU/Linux, Raspbian OS ...) et le plus intéressant pour notre cas est la version RaspianWheezy customiser par MATLAB, qui nous permettra de contrôler et récupérer des données du matériel et des périphériques de la PI à partir du PC par le biais des commandes MATLAB. Ce qui veut dire qu'on peut soit exécuter notre algorithme indépendamment du PC directement sur la carte

Raspberry PI (avec SIMULINK Support Package for Raspberry PI), ou bien utiliser MATLAB support package for Raspberry PI pour acquérir des données à partir de périphériques connectés au PI ; puis analyser, traiter et visualiser ces données dans l'environnement MATLAB du PC (avec DSP System Toolbox) [20]. Cela nous offre un énorme avantage de portabilité comparé à l'Arduino UNO qui se contente (par manque de mémoire interne) de la deuxième spécification.

Deuxièmement, l'existence des GPIO (figure 5.2) et des ports USB permet l'extension de la carte en ajoutant une carte de capteur ou bien une carte d'horloge temps réel. Les plus intéressantes pour notre cas sont soit la carte son USB ou bien la carte DSP [21]. Vu la non disponibilité de la DSP, on a opté pour la carte son USB.

#### **d. Le choix de microphone et haut-parleur**

Les microphones sont sélectionnés pour satisfaire les exigences de faible coût, faible impédance, grand rapport signal sur bruit, non-direction et haute sensibilité. Le haut-parleur est nécessaire pour générer un niveau de pression sonore supérieur au niveau de pression de la source de bruit, avoir une bonne fréquence de réponse aux basses fréquences, avoir une bonne résistance à l'humidité, avoir un faible coût et être compact.

### **5.3 Conception et implémentation**

Pour concevoir ce projet, nous avons dû passer par plusieurs étapes :

1. tout d'abord configurer la connexion entre la Raspberry PI et MATLAB,
2. puis réaliser le programme ANC à travers SIMULINK,
3. et enfin élaborer l'architecture de ce système.

#### **5.3.1 Configuration de la connexion**

La première étape est de télécharger le SIMULINK support package for Raspberry PI, pour ce faire, on clique sur le bouton Add-Ons puis Get Hardware Support Packages (figure 5.7).

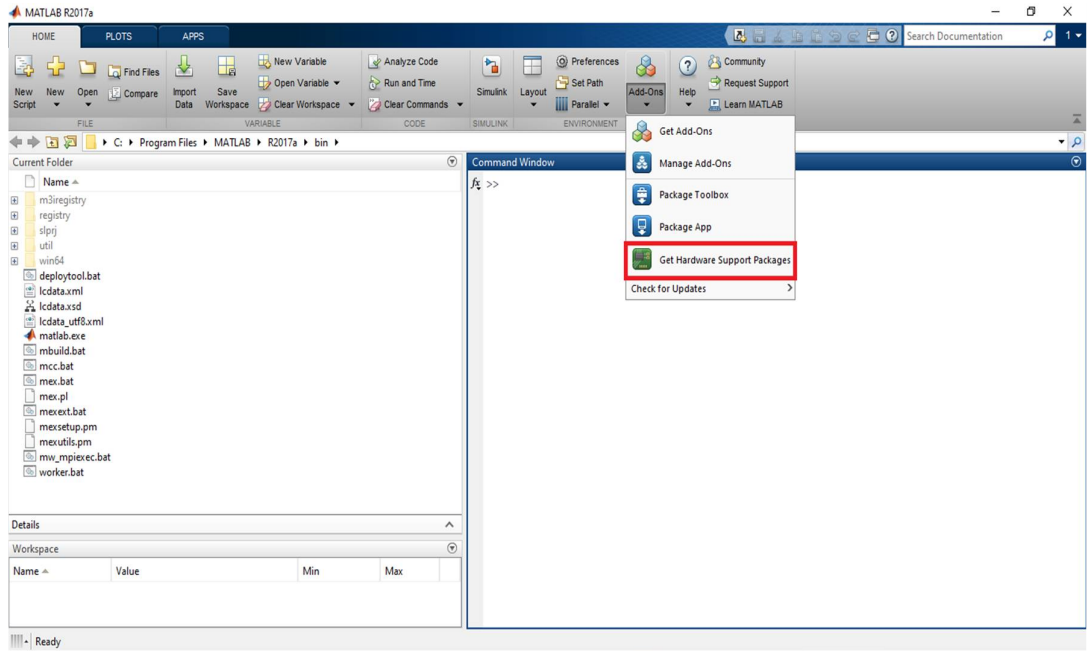


Figure 5.3 : Configuration de la connexion sous MATLAB.

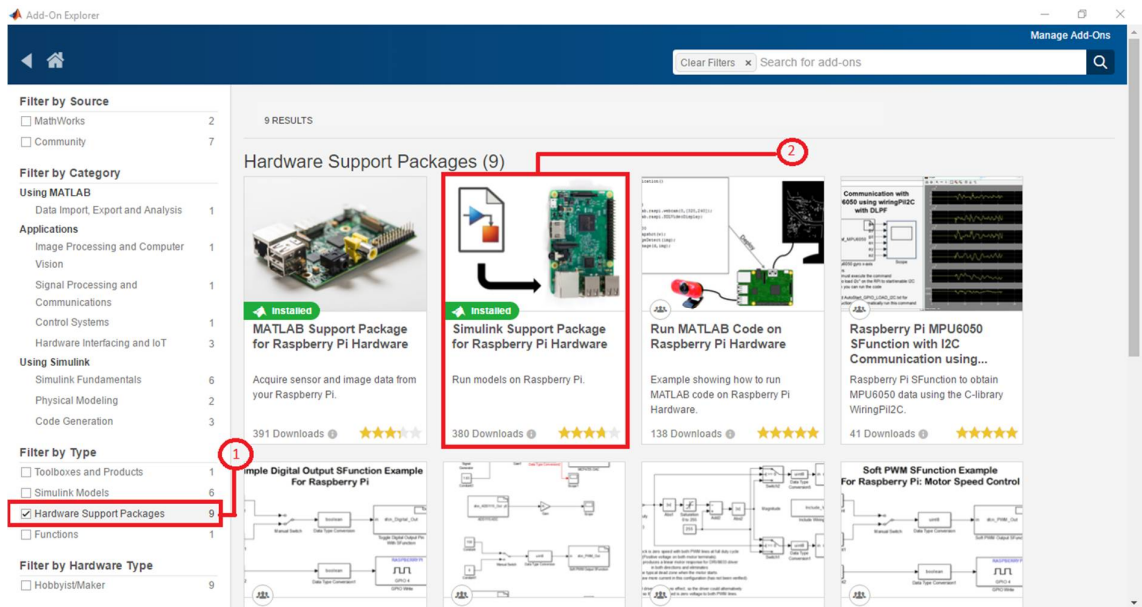


Figure 5.4 : Add-On Explorer.

Ensuite, dans l'onglet « Filter By Type » on choisit Hardware Support Package (c'est là où on retrouve notre SSP comme indiqué sur la figure 5.8).



Maintenant, il reste qu'à suivre les instructions indiquées, mais il faut faire attention au choix du type de carte mémoire utilisé sinon la procédure va s'arrêter à cause de temps d'écriture (il faut utiliser des cartes mémoires de classe 10).

### 5.3.2 Programmmations de l'ANC

Avant de commencer la programmation avec SIMULINK, on doit d'abord configurer notre Raspberry car la carte Raspberry PI ne comporte qu'une seule prise Jack d'où la nécessité d'utiliser une carte son. Ainsi on doit accéder à notre Raspberry et pour cela il suffit juste de se servir soit d'une télévision (avec entre HDMI) et un clavier USB ou bien d'un PC via MATLAB (dans notre projet on a utilisé MATLAB).

Dans la commande Windows on tape le code suivant :

```
>>ras=raspi ('169.254.0.2','pi','raspberrry')
```

```
>>ras.openShell
```

Après cela, une nouvelle fenêtre apparaîtrait (figure 5.9), cette fenêtre représente le terminal de la carte Raspberry, car il ne faut pas oublier que RaspianWheezy (le système d'exploitation fourni par MATLAB) n'est qu'une distribution linux ajustée.

Maintenant il faut accéder au fichier `alsa.conf`, cela peut être fait en utilisant les command suivant figure 5.10 et 5.11 :

```
cd /usr/share/alsa
```

```
nanoalsa.conf
```

```
pi@raspberrypi-dRfnIRPdvs: ~
Using username "pi".

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jan 31 00:24:09 2017 from 169.254.57.241

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

pi@raspberrypi-dRfnIRPdvs:~ $
```

Figure 5.5 : Terminal Raspberry PI.

```
pi@raspberrypi-dRfnIRPdvs: /usr/share/alsa
Using username "pi".

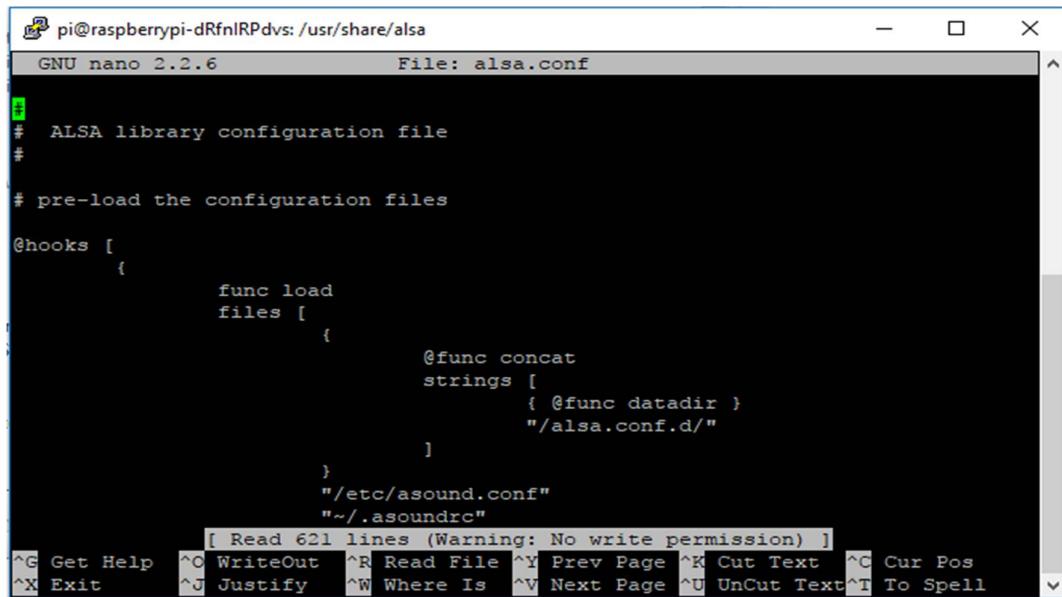
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jan 31 00:24:09 2017 from 169.254.57.241

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

pi@raspberrypi-dRfnIRPdvs:~ $ cd /usr/share/alsa
pi@raspberrypi-dRfnIRPdvs:/usr/share/alsa $ nano alsa.conf
```

Figure 5.6 : Command pour accéder au fichier alsa.conf.



```
pi@raspberrypi-dRfnlRPdvs: /usr/share/alsa
GNU nano 2.2.6 File: alsa.conf
# ALSA library configuration file
#
# pre-load the configuration files
@hooks [
  {
    func load
    files [
      {
        @func concat
        strings [
          { @func datadir }
          "/alsa.conf.d/"
        ]
      }
      "/etc/asound.conf"
      "~/.asoundrc"
    ]
  }
]
[ Read 621 lines (Warning: No write permission) ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

**Figure 5.7 :** Aperçu sur le fichier alsa.conf.

Désormais, on a qu'à remplacer les lignes :

```
defaults.ctl.card 0
defaults.pcm.card 0
```

Par

```
defaults.ctl.card 1
defaults.pcm.card 1
```

Il faut bien faire attention à l'indice de votre carte son, pour le connaître il suffit d'utiliser la commande `aplay -l`.

Après avoir terminé la configuration de la carte son nous allons passer à la programmation de notre ANC via SIMULINK. Pour cela, nous utiliserons la librairie de SIMULINK.

Comme il a été mentionné dans le chapitre 4, la conception du filtre doit passer par deux étapes :

### a. Estimation du chemin secondaire

Comme dans la partie théorique, on s'est servie d'un générateur de bruit coloré pour reproduire le bruit blanc ensuite on le multiplie par un gain pour que le son soit audible (amplifié), et comme le bloc LMS Update ne traite qu'une seule valeur par seconde on a utilisé le bloc Unbuffer. Pour éviter le problème de la boucle on a ajouté un retard aux niveaux de l'erreur.

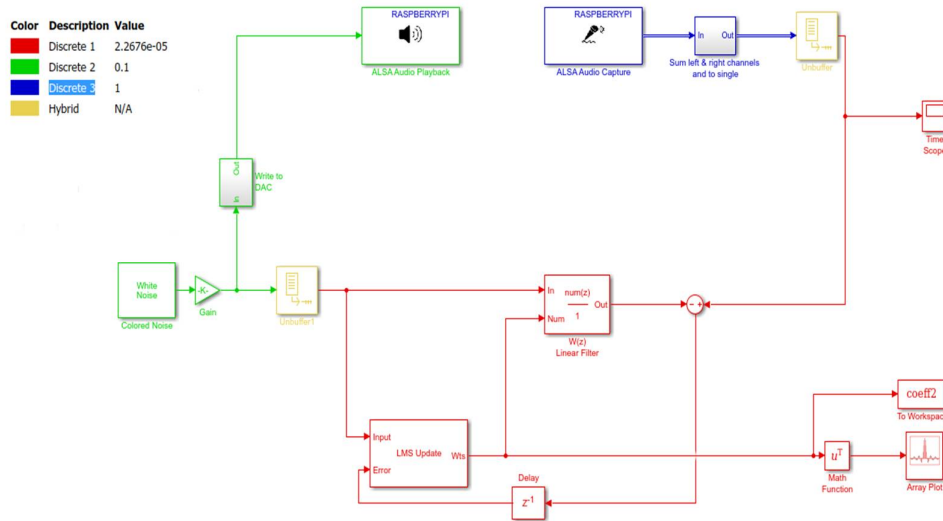


Figure 5.8 : Schéma SIMULINK pour l'estimation de chemin secondaire

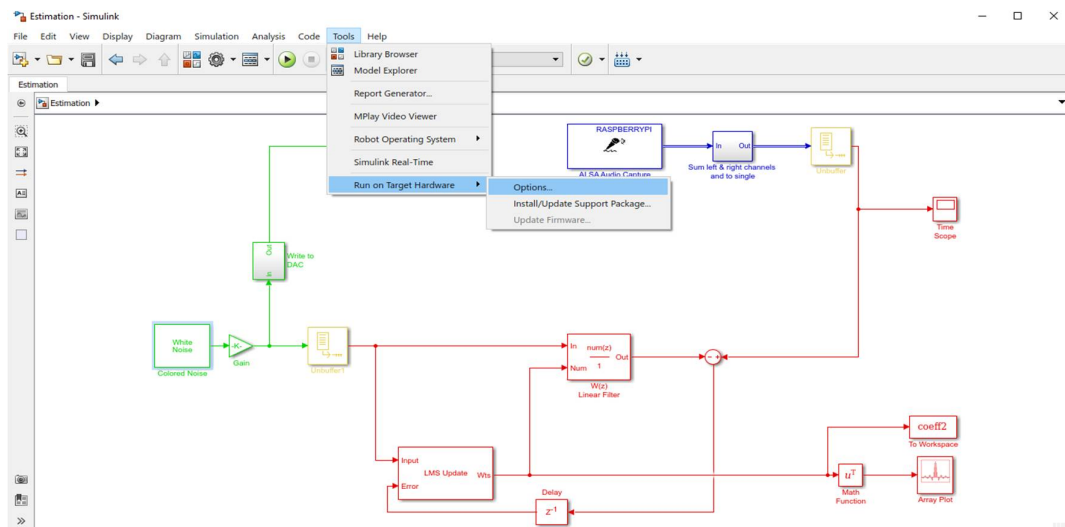
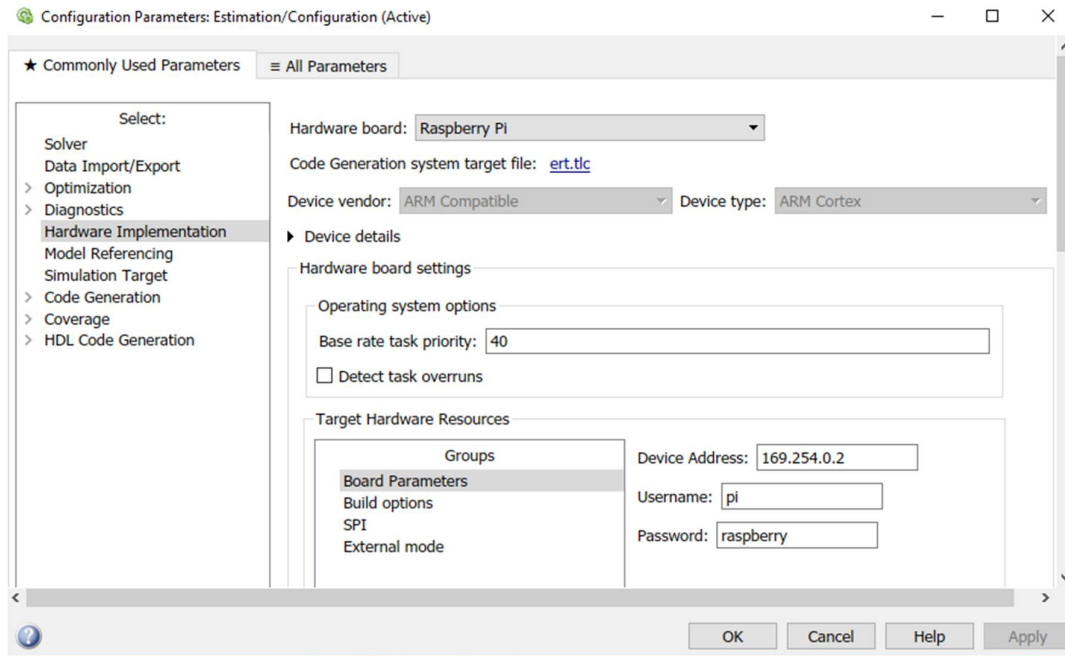


Figure 5.9 : Estimation.

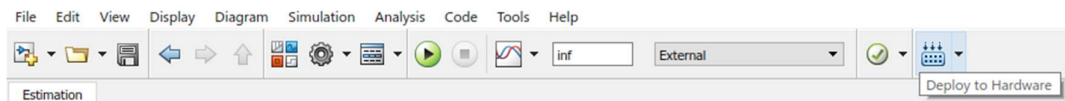
Pour exécuter ce schéma, il faut tout d'abord aller sur « Tools » puis « Run On Target Hardware » et cliquer sur « Options » (figure 5.13).

Là une nouvelle fenêtre apparaît, et on ajustera les paramètres comme il est illustré sur la figure 5.14.



**Figure 5.10 :** Configuration des paramètres.

Après, on n'a qu'à remplacer le temps d'exécution par inf et choisir External. On importe notre projet vers la carte à l'aide du bouton « Deploy to ... » (figure 5.15).



**Figure 5.11 :** Barre de tâches.

### **b. Programme SIMULINK**

Avant de programmer notre ANC, il faut récupérer les coefficients du filtre de l'estimation du chemin secondaire sur le Workspace de MATLAB.

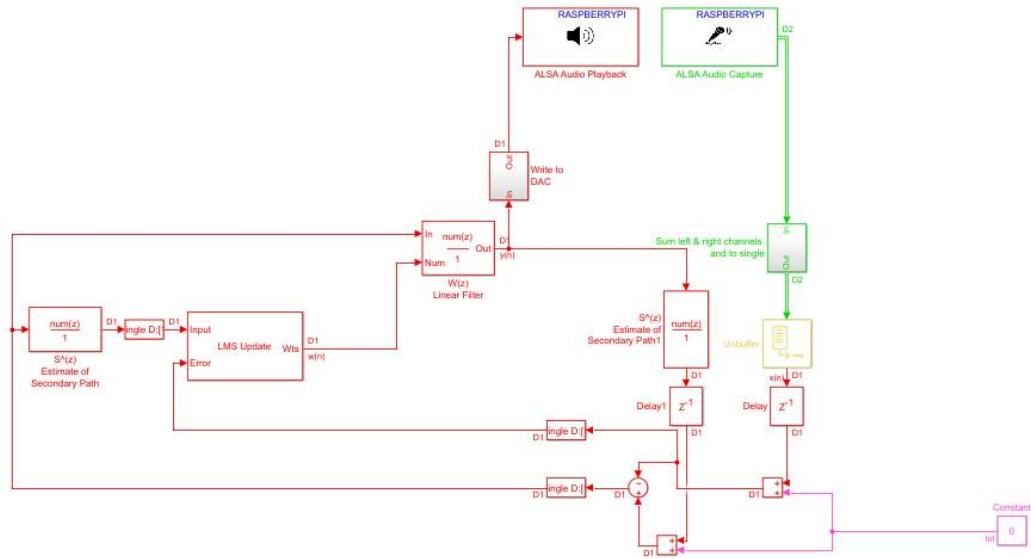


Figure 5.12 : Schéma SIMULINK de l'ANC.

## 5.4 Architecture complète

L'ANC doit compléter l'ensemble de tâches du traitement du signal avant que le bruit primaire n'atteigne le haut-parleur, c'est pour ça que le microphone et le haut-parleur ne doivent pas être ni trop loin ni trop près de la source du bruit.

Pour déterminer la distance optimale du microphone et du haut-parleur, il faut que la condition de causalité soit satisfaite :

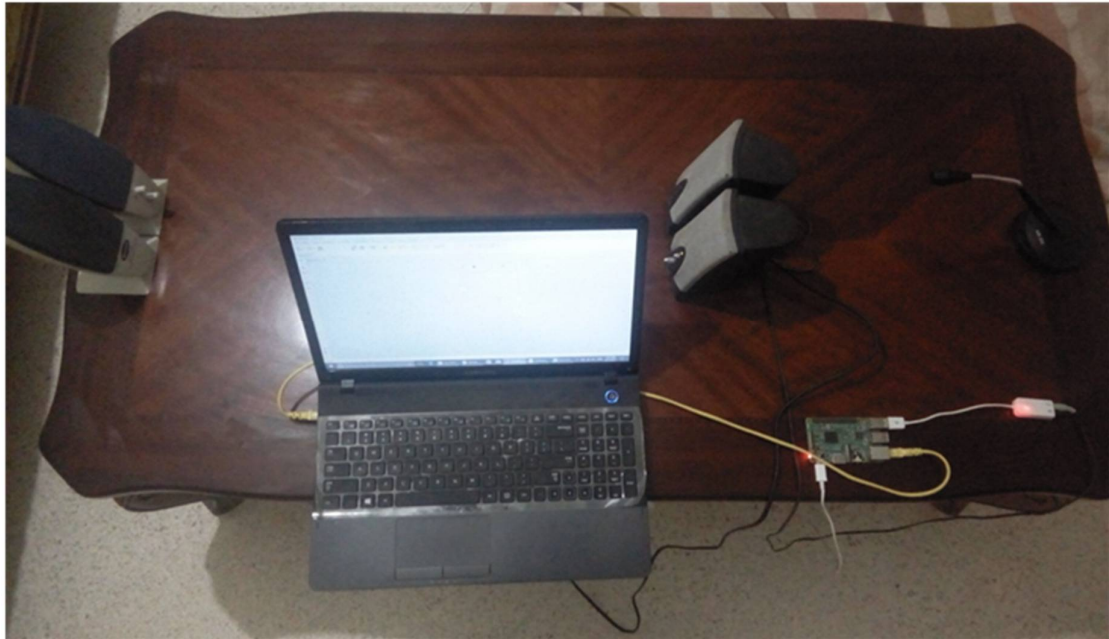
$$\delta_{AB} > \delta_s$$

$$\delta_{AB} = \frac{D_{AB}}{C}$$

$$\delta_s \approx \frac{1}{F_s} + DAC + CAD$$

Où  $\delta_{AB}$  représente le retard acoustique entre un point A et B, et  $\delta_s$  représente le retard du système,  $C$  est la vitesse du son, et  $D_{AB}$  est la distance à respecter.

Dans les deux figures suivantes, nous donnons le prototype du projet pour la réalisation du système ANC feed -back :



**Figure 5.13** : Réalisation de système ANC feed-back avec le PC.



**Figure 5.14** : Réalisation de système ANC sans le PC.

## **5.5 L'application Android**

### **5.5.1 SSH**

SSH est un programme et protocole de communication sécuriser qui permet l'accès à un ordinateur distant (serveur SSH) à partir d'un ou plusieurs appareils (client SSH) en ouvrant une Interface système (shell) sur l'ordinateur distant, habituellement le protocole SSH utilise le port TCP 22 et il peut en plus transférer des fichiers complets (générant un tunnel chiffre entre ces derniers, les informations qui circulent).

Dans notre cas l'application Android est le client et le serveur est la Raspberry PI qui est déjà équipé de cette fonction intégrée dans Linux. Pour cette dernière, il suffit d'activer la fonction en entrant la commande `sudo raspi-config` après cela on active la fonction SSH.

### **5.5.2 Notre application Android**

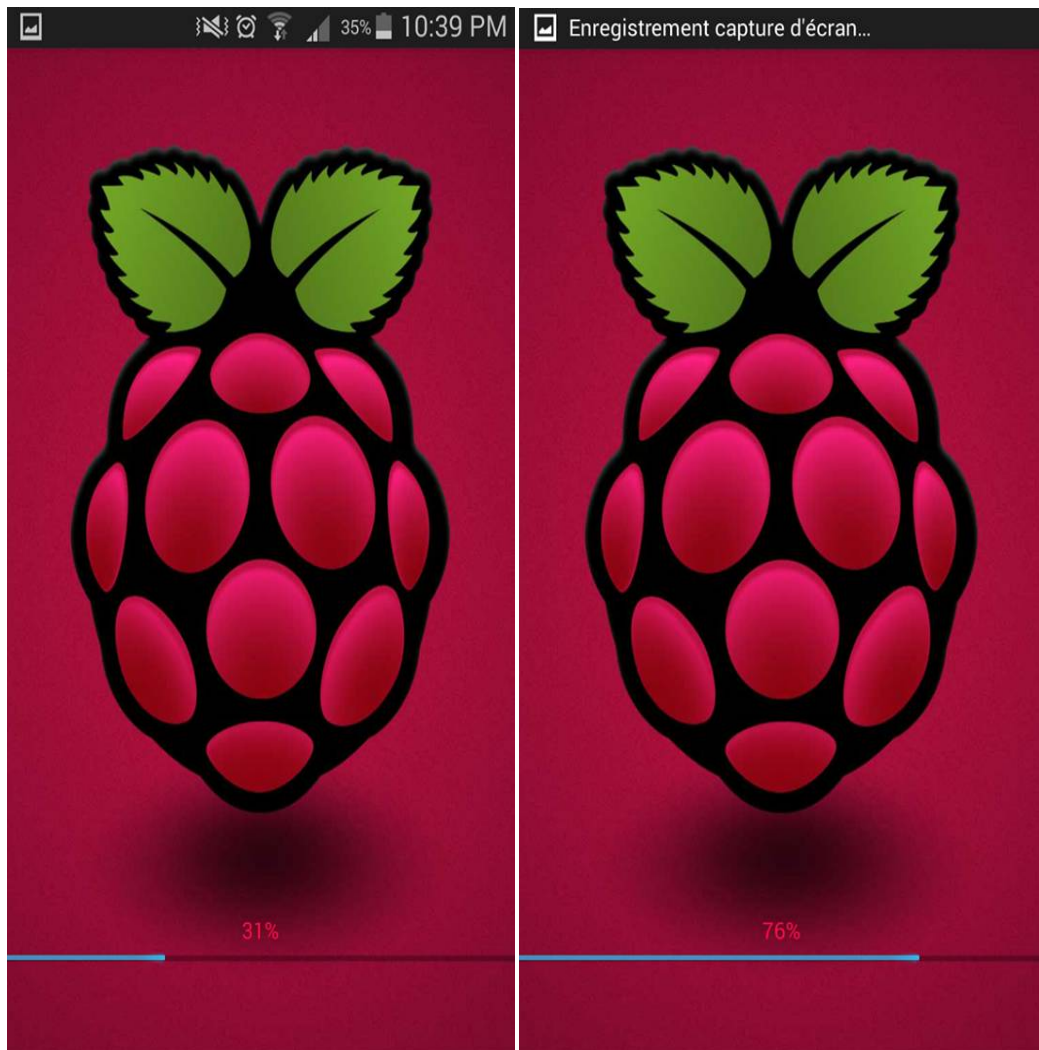
Android est un système d'exploitation le plus populaire pour les appareils mobiles comme les Smartphones et les tablettes. Il est basé sur le célèbre système d'exploitation Linux et il convoite 80% du marché mondiale des Smartphones.

Android offre au développeur d'utiliser plusieurs langages comme JAVA, C++, C (ce qui le rend très riche en matière de bibliothèque). Dans notre projet, les bibliothèques concernées sont les SSH.

Pour faciliter l'utilisation de notre application, nous avons opté pour une interface simple et complète.

Du moment où vous entrez dans l'application, cette dernière essayera de se connecter à Raspberry PI si elle été déjà configurée, sinon un bouton START apparaîtra pour permettre le lancement de l'application dans Raspberry. Ci-après, quatre figures illustreront l'interface de l'application dans un Smartphone :





**Figure 5.15 :** Première interface de l'application (activity\_spalsh\_screen)

Après avoir cliqué sur l'icône de l'application, on entre dans la progresse barre ou la première interface attend quelque secondes pour se connecter.



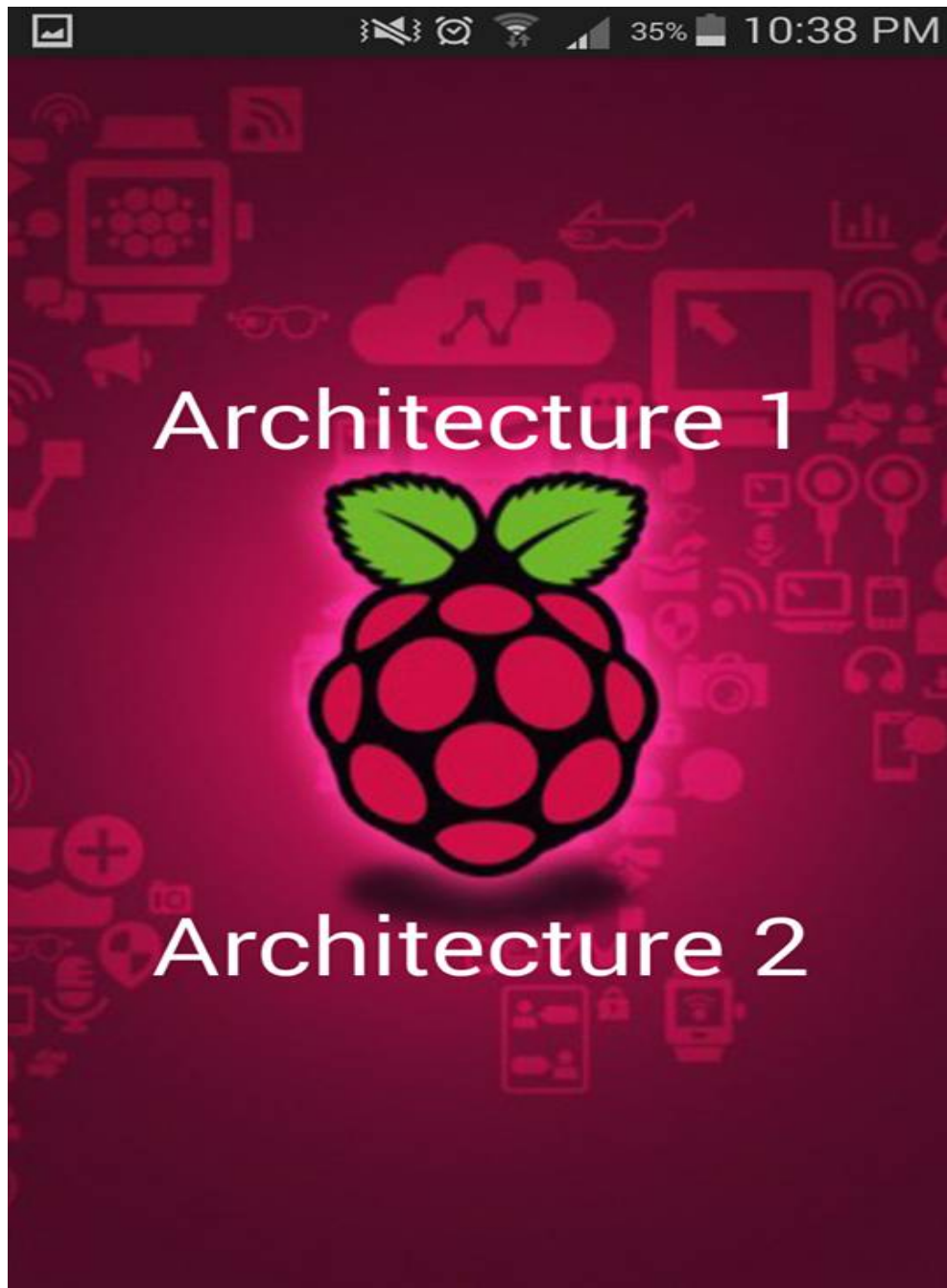
**Figure 5.16** : Deuxième interface (activity\_main).

Dans cette interface, il y'a deux bouton : buttons de Start et bouton de configuration (avant de appuyer sur le buttons Start vous devez configurer les paramètres).



**Figure 5.17** : Troisième interface (activity\_configuration).

Après le clic sur le bouton de configuration, on est obligé de configurer les paramètres de SSH et de les sauvegarder.



**Figure 5.18** : Quatrième interface (activity\_choix).

Dans la dernière interface on peut choisir l'un des deux architectures pour exécuter le code correspondant à cette dernière.

# Conclusion générale

---

Les travaux présentés dans ce projet de fin d'étude ont portés sur la conception d'un système ANC à travers une carte Raspberry PI 3 model B.

Le premier chapitre décrit l'intérêt général des systèmes ANC, et il invoque aussi la motivation du ce projet. Le second chapitre, introduit quelques notions sur l'acoustique afin de bien comprendre les caractéristiques des sons et principalement les bruits. Le troisième chapitre s'intéresse au contrôle actif du bruit en présentant à la fin un état de l'art sur ce dernier. Le quatrième chapitre présente les algorithmes adaptatifs pour le système ANC ainsi que quelques simulations sur ces derniers afin de tester les performances et de choisir la solution la plus appropriée à notre réalisation. Le cinquième et le dernier chapitre décrit pas à pas la procédure suivie pour réaliser notre système, en commençant par la configuration de la carte, puis l'implémentation du système et enfin l'application Android.

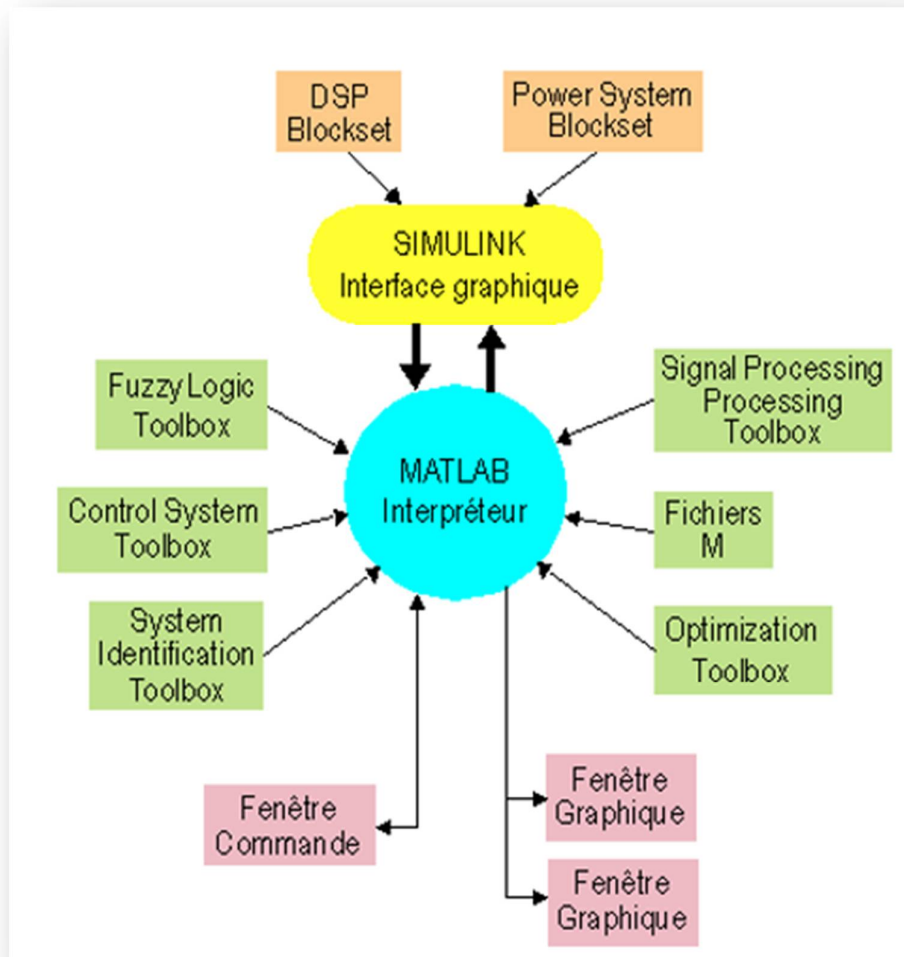
Ce travail nous a permis d'acquérir plusieurs connaissances comme l'intervention de l'informatique dans le domaine électronique, la technologie des systèmes ANC, MATLAB, etc. La manipulation d'une carte Raspberry PI, qui est une tendance actuelle, est une autre connaissance acquise. Nous avons aussi appris l'approche méthodologique de réalisation de projets commençant par l'étude des aspects liés au sujet, passant par la compréhension des besoins et la conception, et arrivant à l'implémentation.

Comme perspectives de ce travail, nous proposons le développement d'autres fonctionnalités plus élaborées comme l'implémentation d'autres algorithmes.

## **A.1 Introduction à MATLAB**

Ce document est un guide simplifié de MATLAB et SIMULINK. Les notions de base sont présentées de façon simple pour permettre aux lecteurs de démarrer rapidement.

MATLAB est un logiciel de calcul matriciel à syntaxe simple. Avec ses fonctions spécialisées, MATLAB peut être aussi considéré comme un langage de programmation adapté pour les problèmes scientifiques. MATLAB est un interpréteur : les instructions sont interprétées et exécutées ligne par ligne. MATLAB fonctionne dans plusieurs environnements tels que X-Windows, Windows, Macintosh. Il existe deux modes de fonctionnement : 1. mode interactif : MATLAB exécute les instructions au fur et à mesure qu'elles sont données par l'utilisateur. 2. mode exécutif : MATLAB exécute ligne par ligne un "fichier M" (programme en langage MATLAB).



**Figure A.1** : Environnement MATLAB

**Fenêtre Commande** : Dans cette fenêtre, l'utilisateur donne les instructions et MATLAB retourne les résultats.

**Fenêtres Graphique** : MATLAB trace les graphiques dans ces fenêtres.

**Fichiers M** : Ce sont des programmes en langage MATLAB (écrits par l'utilisateur).

**Toolboxes**: Ce sont des collections de fichiers M développés pour des domaines d'application spécifiques (Signal Processing Toolbox, System Identification Toolbox, Control System Toolbox, u-Synthesis and Analysis Toolbox, Robust Control Toolbox, Optimization Toolbox, Neural Network Toolbox, Spline Toolbox, Chemometrics Toolbox, Fuzzy Logic Toolbox, etc.)



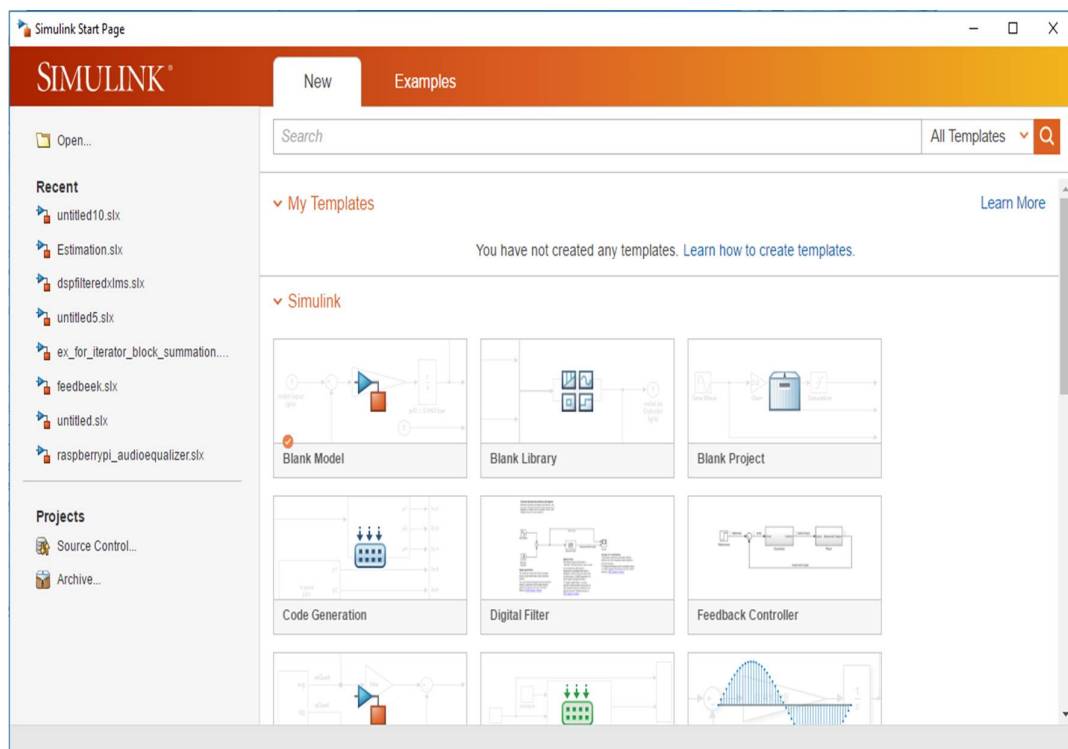
**SIMULINK** : C'est l'extension graphique de MATLAB permettant de travailler avec des diagrammes en blocs.

**Blocksets** : Ce sont des collections de blocs SIMULINK développés pour des domaines d'application spécifiques (DSP Blockset, Power System Blockset, etc.).

## A.2 Introduction à SIMULINK

SIMULINK est l'extension graphique de MATLAB permettant de représenter les fonctions mathématiques et les systèmes sous forme de diagramme en blocs, et de simuler le fonctionnement de ces systèmes.

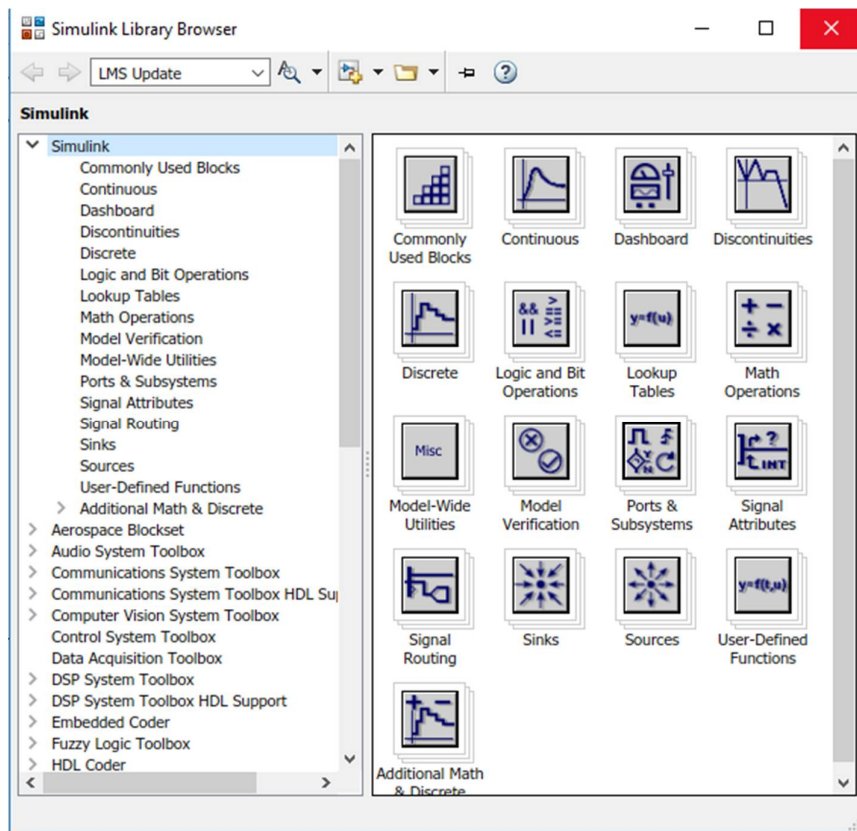
POUR DÉMARRER SIMULINK Dans la fenêtre Commande de MATLAB, taper simulink. La fenêtre Simulink va s'ouvrir



**Figure A.2** : Fenêtre SIMULINK.

Après sa choisir BLANK MODEL puis dans la barre de menu sélectionner Tools en suit appuie sur LIBRARY BROWSER





**Figure A.3 :** Navigateur de la librairie SIMULINK.

Cette fenêtre contient des collections de blocs que l'on peut ouvrir en cliquant (double) dessus :

Sources Sources de signaux

Discrete Blocs discrets

Linear Blocs linéaires

Nonlinear Blocs nonlinéaires

Connections Entrée/sortie, multiplexeur/démultiplexeur, etc.

Demos Démonstrations

Blocksets & Toolboxes Blocksets et toolboxes

## **CONSTRUCTION D'UN DIAGRAMME SIMULINK**

Ouvrir les collections de blocs en cliquant dessus (double). Faire glisser dans la fenêtre de travail les blocs dont on a besoin pour construire le diagramme. Faire des liaisons entre les blocs à l'aide de la souris. Lorsqu'on clique (double) sur un bloc, une fenêtre de dialogue s'ouvrira. On peut alors changer les paramètres de ce bloc. Une fois terminé, on ferme la fenêtre de dialogue. Une fois le diagramme terminé, on peut l'enregistrer dans un fichier : dans le menu File, choisir Save As et donner un nom (\*.mdl) au fichier.

### **A.3 Introduction a Android studio**

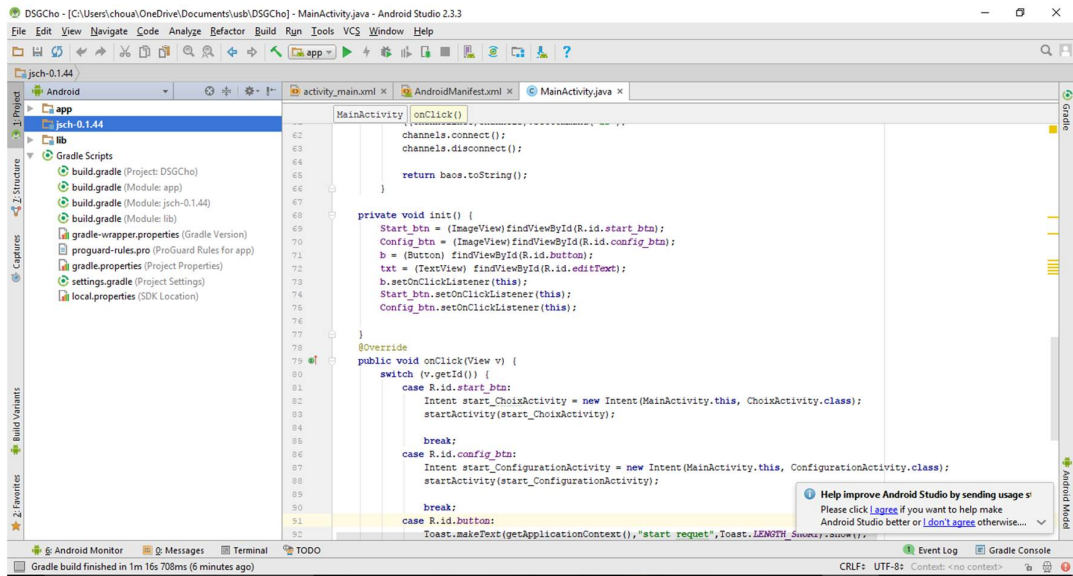
Android Studio est l'environnement de développement officielle de Google pour développer des applications Android, il se comporte principalement d'un SDK, Emulateur, Editeur :

➤ Le SDK :

Le Software développement kit proposer par Google comporte un débogueur des bibliothèques logicielles, de la documentation, des exemples de code bref tous se qu'un développeur aura besoin.

➤ L'émulateur :

L'émulateur et logicielle qui permet de simule un vrai smartphone Android avec tous ses fonction, ce qui simplifie largement la tâche du développement.





# Bibliographie

---

- [1] M. S. R. M. Moazzam, «Performance Evaluation of Different Active Noise Control (ANC) Algorithms for Attenuating Noise in a Duct,» Blekinge Institute of Technology, Karlskrona, Sweden, 2013.
- [2] I. P. Sen M. Kuo, Design of Active Noise Control Systems With the TMS320 Family, texas: Digital Signal Processing Products—Semiconductor Group, 1996.
- [3] S. Johansson, Active Control of Propeller Induced Noise in Aircraft, Ronneby, Sweden, dec 2000.
- [4] P. Lueg, «Process of Silencing Sound Oscillations». usa Brevet 2,043,416, juin 1996.
- [5] C. H. Hansen, «FUNDAMENTALS OF ACOUSTICS,» Department of Mechanical Engineering, University of Adelaide, Australia, 1994.
- [6] C. Woodford, 21 juin 2016. [En ligne]. Available: <http://www.explainthatstuff.com/sound.html>. [Accès le 5 juin 2017].
- [7] Z. N. Abidat, «conception d'un système anti-bruit pour des systèmes de ventilation (partie softward),» Université de montpellier 2, montpellier , sep, 2006.
- [8] A. M. a. U. Schaufelberger, Active Noise Control, Rapperswil-Jona: University of Applied Sciences Rapperswil HSR, 2005.
- [9] D. Mercier, le livre des techniques du son, Dunod, 2002.
- [10] S. M. K. a. D. R. Morgan, «"Active Noise Control Systems.",» John Wiley & Sons, 1996.
- [11] L.-C. C. T. M. Kuang-Hung liu, «Active Noise Cancellation project,» 2008.
- [12] D. R. M. Sen M. Kuo, «Active Noise Control: A Tutorial Review ,Proc IEEE,» vol. 87, n° %1no.6, 6 juin 1999.
- [13] B. T. Sen .M. Kuo, «Real Time Digital Signal Processing,» england, John Wiley & Sons Ltd, 2006.

- [14] H. F. E. G. M. Olson, «"Electronic Sound Absorber,"», vol. 25, Nov 1953.
- [15] H. F. Olson, «"Electronic Control of Noise, Vibration, and Reverberation",» vol. 28, 1956.
- [16] D. R.Chinaboina, «Adaptif ALGORITHMS FOR ACOUSTIC ECHO CANCELLATION IN SPEECH PROCESSING,» *International Journal of Recent Research and Applied Studies*, 2011.
- [17] B. W. Max Kamenetsky, «A Variable Leaky LMS Adaptive Algorithm,» ISL, Department of Electrical Engineering, Stanford University, Stanford CA, USA.
- [18] P. P. H. Bageshree Pathak, «seekdl,» [En ligne]. Available: <http://www.seekdl.org>.
- [19] raspberrypi, «[www.raspberrypi.org](http://www.raspberrypi.org),» 2016. [En ligne]. Available: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>.
- [20] mathworks, «Raspberry Pi Programming with MATLAB and Simulink,» 2017. [En ligne]. Available: <https://www.mathworks.com/discovery/raspberry-pi-programming-matlab-simulink.html>.
- [21] crazy-audio, «<http://www.crazy-audio.com>,» 2014. [En ligne]. Available: <http://www.crazy-audio.com/projects/hifiberry-dsp-light/>.
- [22] R. S. Orlando J. Tobias, «PERFORMANCE COMPARISON OF THE FXLMS, NONLINEAR FXLMS AND LEAKY FXLMS ALGORITHMS IN NONLINEAR ACTIVE CONTROL APPLICATIONS,» LINSE - Circuits and Signal Processing - Florianópolis - SC - Brazil.
- [23] Gotron Electronica, «[www.gotron.be](http://www.gotron.be),» [En ligne]. Available: <https://www.pinterest.com/pin/515873332298992803/>.
- [24] element14, «[www.element14.com](http://www.element14.com),» [En ligne]. Available: <https://www.element14.com/community/docs/DOC-73950/1/raspberry-pi-3-model-b-gpio-40-pin-block-pinout>.