國 立 交 通 大 學
電機與控制工程研究所

碩 士 論 文

應用於個人電腦環境之即時語音純化系統設計

Real-time Speech Signal Purification System Design for Desktop Environment

研 究 生： 康 創 閔

指導教授： 胡 竹 生 博士

中 華 民 國 九 十 三 年 七 月

應用於個人電腦環境之即時語音純化系統設計

Real-time Speech Signal Purification System Design for Desktop Environment

研 究 生：康　創　閔　　Student：Edison Kang
指導教授：胡　竹　生　博士　Advisor：Jwu-Sheng Hu

國立交通大學
電機與控制工程學系
碩 士 論 文

A Thesis
Submitted to Institute of Electrical and Control Engineering
College of Electrical Engineering and Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of Master
in

Electrical and Control Engineering

July 2004

Hsinchu, Taiwan, Republic of China

中 華 民 國 九 十 三 年 七 月

博碩士論文授權書

(國科會科學技術資料中心版本，93.6.17)

本授權書所授權之論文為本人在 國立交通 大學 電機與控制工程 系所 ＿＿＿＿＿＿ 組 九十二 學年度第 二 學期取得 碩士 學位之論文。

論文名稱：應用於個人電腦環境之即時語音純化系統設計

Real-time Speech Signal Purification System Design for Desktop Environment

V **同意** □ **不同意** **(政府機關重製上網)**

本人具有著作財產權之論文全文資料，授予行政院國家科學委員會科學技術資料中心、國家圖書館及本人畢業學校圖書館，得不限地域、時間與次數以微縮、光碟或數位元化等各種方式重製後散佈發行或上載網路。

本論文為本人向經濟部智慧財產局申請專利（未申請者本條款請不予理會）的附件之一，申請文號為：＿＿＿＿＿＿，註明文號者請將全文資料延後半年後再公開。

V **同意** □ **不同意** **(圖書館影印)**

本人具有著作財產權之論文全文資料，授予教育部指定送繳之圖書館及本人畢業學校圖書館，為學術研究之目的以各種方法重製，或為上述目的再授權他人以各種方法重製，不限地域與時間，惟每人以一份為限。

上述授權內容均無須訂立讓與及授權契約書。依本授權之發行權為非專屬性發行權利。依本授權所為之收錄、重製、發行及學術研發利用均為無償。上述同意與不同意之欄位若未鉤選，本人同意視同授權。

指導教授姓名： 胡竹生 教授

研究生簽名： 康創閔 學號：9112591

（親筆正楷） （務必填寫）

日期：民國 93 年 7 月 日

---

1. 本授權書 (得自http://sticnet.stic.gov.tw/sticweb/html/theses/authorize.html 下載或至 http://www.stic.gov.tw/首頁右下方下載) 請以黑筆撰寫並影印裝訂於書名頁之次頁。
2. 授權第一項者，請確認學校是否代收，若無者，請個別再寄論文一本至臺北市(106-36) 和平東路二段 106 號 1702 室 國科會科學技術資料中心 黃善平小姐。 (電話:02-27377606 傳真：02-27377689)
3. 本授權書已民國 85 年 4 月 10 日送請內政部著作權委員會（現為經濟部智慧財產局）修正定稿，89.11.21 部份修正。
4. 本案依據教育部國家圖書館 85.4.19 台(85)圖編字第 712 號函辦理。

國立交通大學

論文口試委員會審定書

本　　校 電機與控制工程學系碩士班 康創閔君

所提論文 應用於個人電腦環境之即時語音純化系統設計
Real-time Speech Signal Purification System Design
for Desktop Environment

合於碩士論文資格水準、業經本委員會評審認可。

口試委員：

指導教授：

系主任：　　教授

中華民國　九十三 年 七 月　　日

應用於個人電腦環境之即時語音純化系統設計

研究生：康　創　閔　　指導教授：胡　竹　生　博士

國立交通大學電機與控制工程研究所碩士班

摘　要

本論文針對桌面環境的訊號的干擾源：如喇叭、迴響環境，使用適應性麥克風陣列訊號處理，抑制干擾源的影響，並使輸入輸出即時化。本論文亦提出一個符合實驗室規模、使用 USB1.1 介面、8 通道之即時性麥克風陣列訊號處理實驗平臺，已實作完成。以此實驗平臺錄製麥克風陣列語音樣本供相關研究，並實作完成適應性空間濾波器。

論文主要分三大部分，第一部分介紹陣列式麥克風波束形成(Beamforming)演算法的概念、適應訊號處理理論。第二部分介紹實驗平臺的架構。第三部分則是演算法在實驗平臺上的實作，實際輸入輸出的結果與分析。

Real-time Speech Signal Purification System Design for Desktop Environment

Student： Edison Kang Advisor： Prof. Jwu-Sheng Hu

Institute of Electrical and Control Engineering

ABSTRACT

The objective of the thesis is to subpress the interference sources of the desktop environment like loudspeaker, echo, and environmental noise by using adaptive microphone array. The implementation also requires a real-time input and output of signals. A flexible real-time array-processing platform of 8 sensors using USB 1.1 interface is proposed and implemented. Using this platform, an adaptive spatial filter is also implemented.

The thesis is divided into three parts. The first part is an introduction of the microphone array beamforming algorithm and adaptive filter used. The second part is the platform implementation. The third part is implementation of the adaptive microphone array algorithm on the platform and discussions of the experimental results.

誌謝

對於本論文的完成，首先感謝我的指導教授胡竹生博士。老師不僅在學識上使我獲益匪淺，教導我做學問需有的態度和方法，及正確的研發觀念，亦耳濡目染中的從他身上學習到待人處事的方法，對於老師多年來的教導，誠摯的致上最真誠的謝意。

另外，感謝在苦悶的研究生生活裡，一直支持著我的一些好伙伴：感謝伯彥學長與立偉學長，在讓硬體上提供了許多相關的知識及建議；感謝价呈兄帶我們去南寮吃生猛海鮮, 讓大家有力氣及精神做好研究工作;感謝維瀚兄成立了 XLAB 重訓組，讓實驗室成員都擁有健康的體魄；感謝發願吃素的新好男人宗敏，以後從大甲回來的時候要記得帶芋頭酥；感謝家瑋帶我們去室內棒球場打棒球，在亞錦賽及奧運棒球賽過後，可以持續感覺到身體裡的熱血；感謝常幫大家訂便當的認真的憶如，即使妳已經從實驗室一姊退休了，妳仍是我心目中永遠的一姊；感謝手藝好的安喬兄，讓我們在實驗室可以吃到好吃的熱騰騰的鮪魚三明治跟火鍋；感謝了解我的幽默的順子，在我講冷笑話的時候還是笑的很大聲；感謝穿藍白拖鞋的阿毛，讓我不必獨自承受一個人穿藍白拖鞋的孤單；感謝差點去澳洲留學的俊德兄，讓實驗室充滿在袋鼠與無尾熊的幻想中；感謝會煮咖啡的鏗元，讓我了解到好喝的咖啡與不好喝的咖啡之間的差異；及橫渡日月潭的士奇、求知欲很強的晏榮、玩小企鵝很厲害的群棋、實驗室快過期的一姊岑思、開雙 B 超有義氣的興哥。感謝你們在我研究所的生涯中給我許多的歡樂與幫忙，使我在這兩年間留下了美好的回憶。

最後我要感謝家人對我的關心和照顧，因為有你們，我才能順利完成學業，最後，在此感謝每一位關心我的人，謝謝你們。

目　錄

表　列

圖　列

第一章 緒論

1.1 研究動機

桌面環境通常是吵雜的，往往存在多個訊號干擾源，如喇叭、電腦風扇、密閉空間反射，都會影響到語音訊號的輸入，當語音訊號受到干擾源影響時，辨識率會大為降低。為了純化語音訊號，需要一個語音輸入介面，去除掉環境中干擾源的影響。

以往要消除干擾源的影響，往往要知道干擾源的特性，一般的處理方式是由時域、頻域去分析干擾源的特性，如果目前的干擾源在時域、頻域上沒有什麼固定的特徵，如電腦不斷的播放不同的音樂，那麼藉由時域、頻域去消除干擾源的影響，效能往往不理想。在論文中我們擷取了訊號在空間上的特徵來做處理。桌面環境下的干擾源位置經常是固定的，如喇叭放好之後就很少再會去變動喇叭的位置，只要針對當時的環境做空間濾波(Spatial Filter)，由空間上的特徵來對訊號處理，如此即可將與使用者不同角度的干擾源影響去除掉，達到增加語音訊號的訊噪比(SNR)，如此即可純化語音訊號，提高語音訊號辨識率。

1.2 研究目標

設計空間濾波器，首先要有將陣列式訊號即時性的傳回電腦的設備，因此需要架構出麥克風陣列的系統，經由 USB 介面傳回電腦。

再來，模擬各種空間濾波的設計方法，找出效能好的演算法，並針對目前的桌面環境、考慮陣列式麥克風系統的誤差，選定一種最好的演算法，加以實現。

訊號即時性的輸入，經由演算法的處理之後，做即時性的輸出。

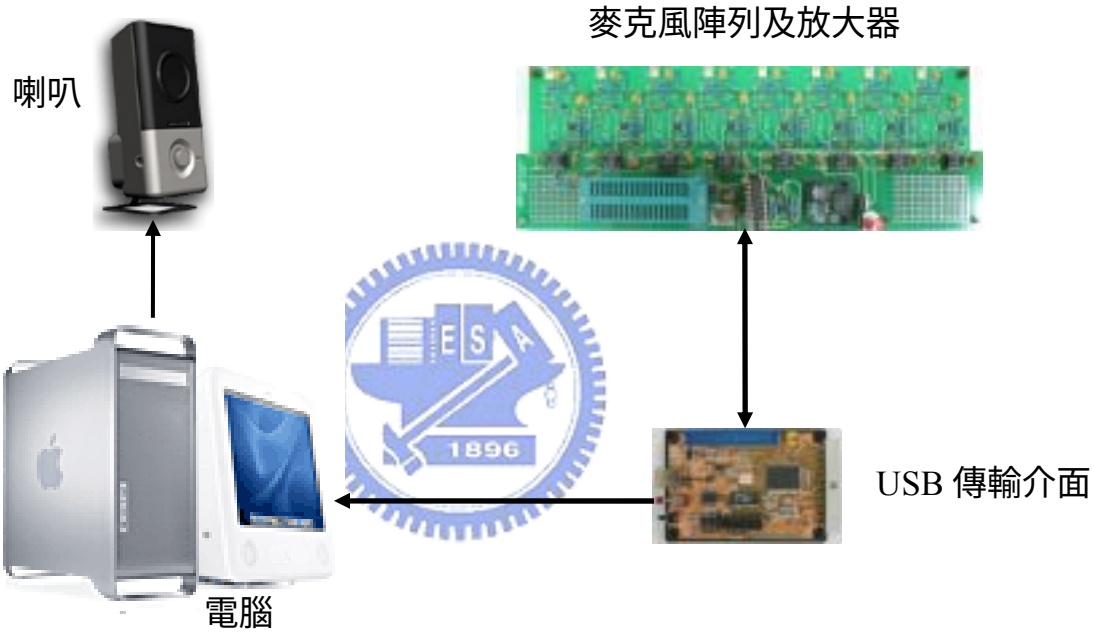

圖 1. 系統架構簡圖

1.3 序言

這篇論文包含了三個主要的部分，分別是即時性演算法的理論、實驗平臺的架構與實現與即時性演算法的驗證。底下將大致描述三個主要部分的內容：

第二章：敘述演算法的理論，包含三個部分

1、 陣列式訊號處理：陣列式訊號處理特性介紹、波束形成、波長與感應元件的限制、理想均勻線性陣列模型非理想均勻線性陣列模型。

2、 適應性訊號處理：適應性訊號處理特性介紹、基本架構、LMS、NLMS。

3、 適應性陣列式訊號處理：綜合 1、2 的特性、設計適應性空間濾波器。

第三章：實驗平臺架構

描述實驗平臺的硬體架構，傳輸介面、在硬體上的韌體如何運作，在電腦上的驅動程式與軟體如何運作。

第四章：演算法在實驗平臺上的驗證

理想上的結果與實際處理的結果的分析。

第五章：結論

第二章 適應性陣列訊號處理

2.1 陣列式訊號處理

2.1.1 陣列式訊號處理簡介

陣列式訊號，是指數個相同的感應器排成特定的形狀，接收空間中傳遞的訊號，經過處理之後，達到空間濾波(Spatial Filter)的功能。空間濾波是指訊號在空間中從不同的角度的輸入，分別給予不同的增益。當不同的訊號在頻譜上重疊的部分太多，如語音訊號與 while noise，一般的頻域濾波器並無法將兩個訊號分開，但如果語音訊號與雜訊源對於陣列式感應器的輸入方向不同，具有不同的空間上的資訊，即可利用這樣的資訊設計空間濾波器，將不需要的訊號濾除，達到只接收特定方向的訊號。

2.1.2 波束形成(Beamforming)

波束形成的功能是了接收空間中特定角度的訊號，同時消減其他方向的干擾。典型的 MVDR 波束形成圖如下：

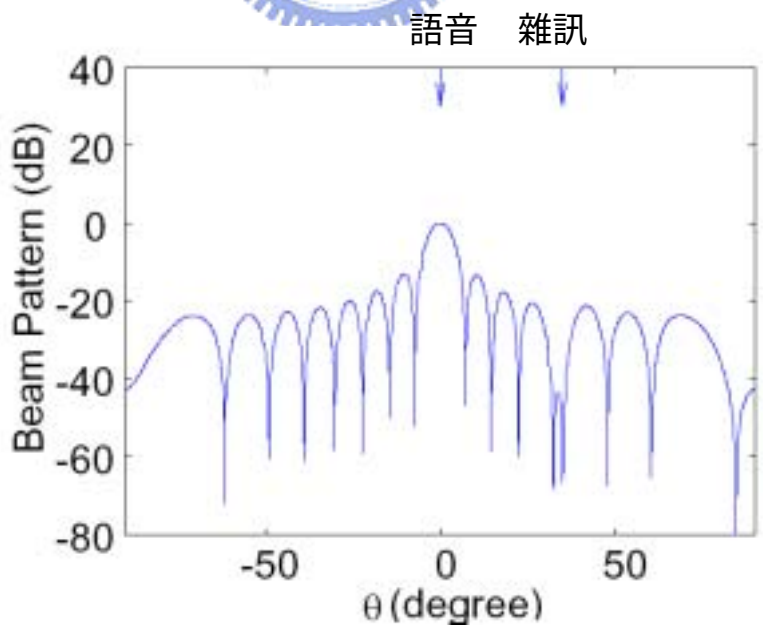

圖 2. 典型 MVDR 波束型成圖

利用輸入訊號具有空間上的資訊，將語音訊號輸入方向的增益固定在 0dB，其他方向的增益被壓低在-10dB 以下，特別是雜訊輸入方向的增益被壓低到-40dB。如此當輸入訊號經過處理之後，只會保留住語音訊號，其他方向的訊號變小，特別是雜訊輸入方向的訊號會變的更小。輸入輸出的 SNR 即可大為提高。

2.1.3 理想均勻線性陣列模型

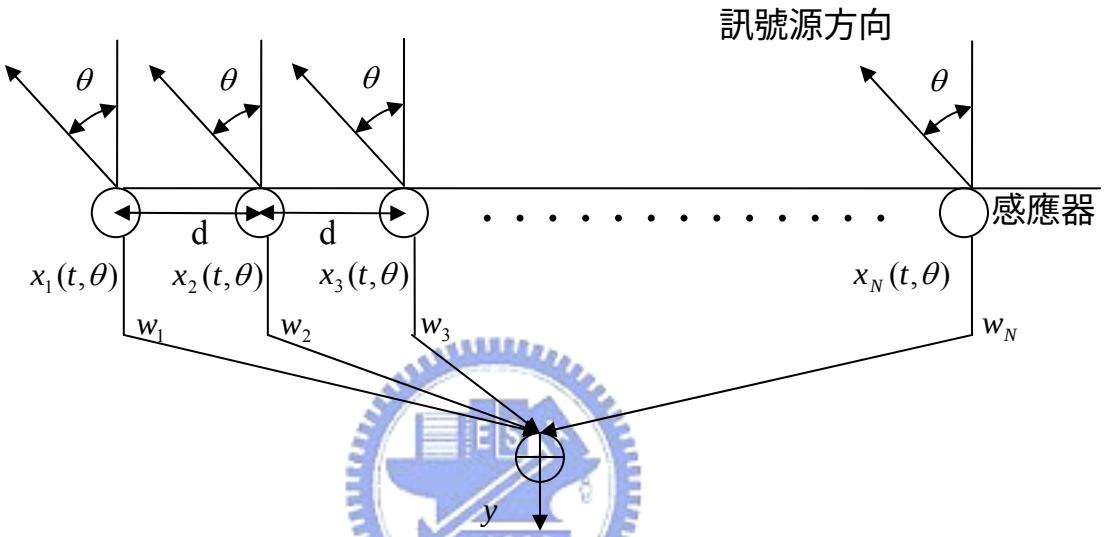

圖 3. 理想均勻線性陣列模型

上圖所示為均勻線性陣列模型的排列方式，是一般最常用的模型。訊號源設定為遠場平面波，訊號平面波在空間中傳播從 x_1 到 x_N 之間的能量損失忽略不計，所以每個感應器對於訊號源的相對角度均為 θ。若訊號傳遞速度為 c，第 1 個感應器收到的訊號 $x_1(t,\theta)$ 為 $s(t,\theta)$，那麼第 N 個感應器收到的訊號 $x_N(t,\theta)$ 就會是 $s(t-(N-1)\times d\times\sin\theta/c,\theta)$，即為第 1 個感應器收到的訊號延遲 $(N-1)\times d\times\sin\theta/c$ 時間。每個感應器接收到的訊號有不同的延遲，這些不同的延遲即是陣列式訊號處理中最重要的空間上的資訊。

當陣列式訊號環境滿足下列的假設：

1、 窄頻訊號(Narrow Band)。

2、　遠場平面波(Far Field Plane Wave)。

輸出訊號 y 就可以表示為：

$$y = \sum_{i=1}^{N} w_i x_i(t,\theta) \qquad (2\text{-}1\text{-}1)$$

$$x(t,\theta) = \begin{bmatrix} x_1(t,\theta) \\ x_2(t,\theta) \\ \mathrm{M} \\ x_N(t,\theta) \end{bmatrix} = s(t,\theta) \begin{bmatrix} 1 \\ \lambda^{jk_c d\sin\theta} \\ \mathrm{M} \\ \lambda^{jk_c (N-1) d\sin\theta} \end{bmatrix} \qquad (2\text{-}1\text{-}2)$$

$k_c = \dfrac{2\pi}{\lambda_c}$ ，　λ_c 窄頻訊號載波波長。

理想狀況之下列出的式子非常的具有規律性，大部分的波束形成演算法都是在這樣的基礎之下推導出有規則的數學式，而方便分析與使用。

2.1.4 非理想均勻線性陣列模型

理想狀況之下推導出來的式子，無法應用在語音訊號的環境中，在語音輸入訊號上，使用者與麥克風的距離是相當近的，一般情況介於 0.1~1m 之間，如下圖所示：

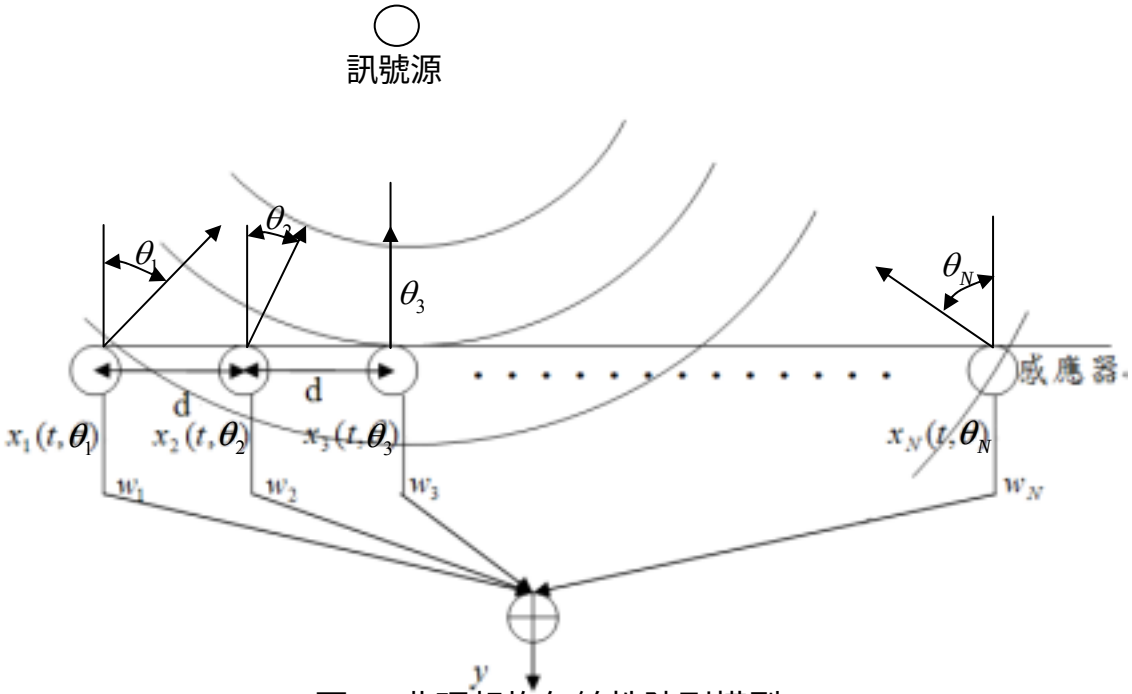

圖 4. 非理想均勻線性陣列模型

當使用者在近距離使用麥克風錄音的時候，屬於近場(Near Field)球形波，語音訊號亦非窄頻訊號，而每個麥克風與放大電路的特性都不大相同，造成的影響主要有下列幾項：

1、 元件不匹配：每個麥克風對於頻率ω與方向θ的增益不同，放大電路也因為元件本身有誤差而造成頻率響應不同，因此收到的訊號要再乘以各別的麥克風與放大電路的頻率響應與方向響應$g_i(\omega,\theta_i)$。

2、 麥克風指向訊號源的方向：每一個麥克風指向訊號源的角度均不同，不再是單一的θ，因此式子並無法化簡成只有 t、θ兩個變數。

3、 能量散失無法忽略：因為距離非常接近，一般情況介於 0.1~1m 之間，因此球形波達到第 1 個麥克風與到第 N 個麥克風的能量逸散相當嚴重，以下面的情況為例子：

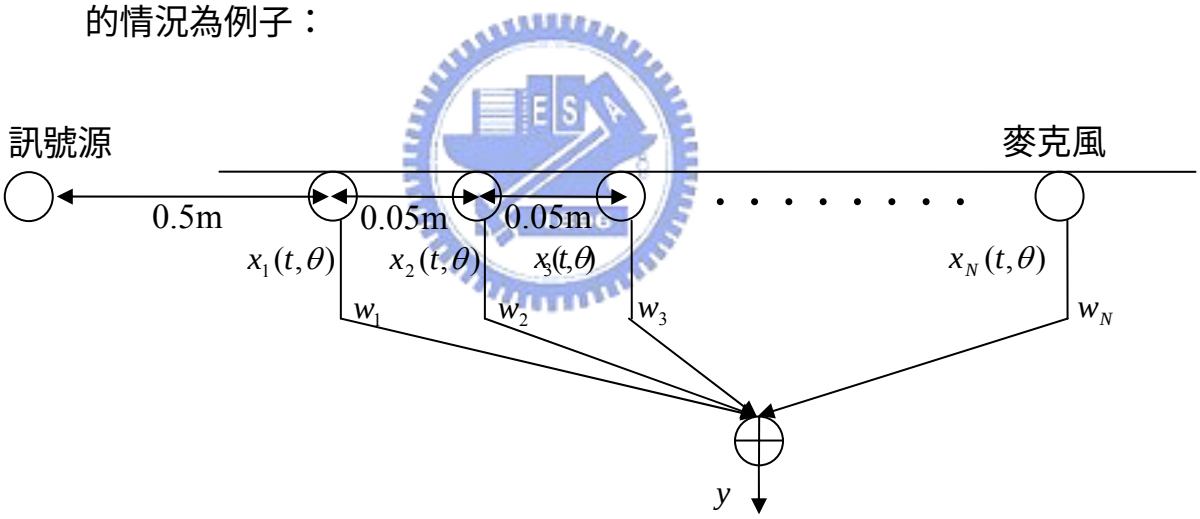

圖 5. 近場效應能量逸散

如圖所示，訊號源與所有麥克風在同一直線上，且在所有麥克風的左邊，距離第 1 個麥克風為 0.5m，每個麥克風間距為 0.05m，陣列式麥克風個數為 8 個，語音訊號能量集中在球型波的表面。

則第 1 個麥克風收到的能量:第 8 個麥克風收到的能量

$$= \frac{1}{(0.5)^2} : \frac{1}{(0.5+7\times 0.05)^2} = 2.89:1$$

能量逸失高達 2.89 倍，無法將這樣的效應忽略不計。

將每個麥克風各別的能量逸失對增益造成的影響表示 k_i。

所以近場寬頻的環境下，陣列式麥克風處理完的訊號只能以下面的式子表示

$$y = \sum_{i=1}^{N} w_i x_i(t,\theta_i) g(\omega,\theta_i) k_i \qquad (2\text{-}1\text{-}3)$$

式子化簡將會變得相當的困難，在這樣子的基礎之下，大部分的波束形成演算法，也會變得相當複雜，分析與使用將變得相當不容易。

2.2 適應性訊號處理

2.2.1 適應性訊號處理簡介

一般而言，設計濾波器往往需要知道設計的濾波器的特徵，才能做濾波器的設計，如低通濾波器需要知道截止頻率，頻寬，增益。而適應性濾波器則不需要知道濾波器的特徵，只要根據輸出訊號與希望達成的訊號造成的誤差訊號，配合適應訊號演算法，即可改變濾波器特性，以完成某些特定的需要。而適應性濾波器的特性改變是自動的，並不需要使用者實際干涉調整的過程。

2.2.2 適應訊號處理基本架構

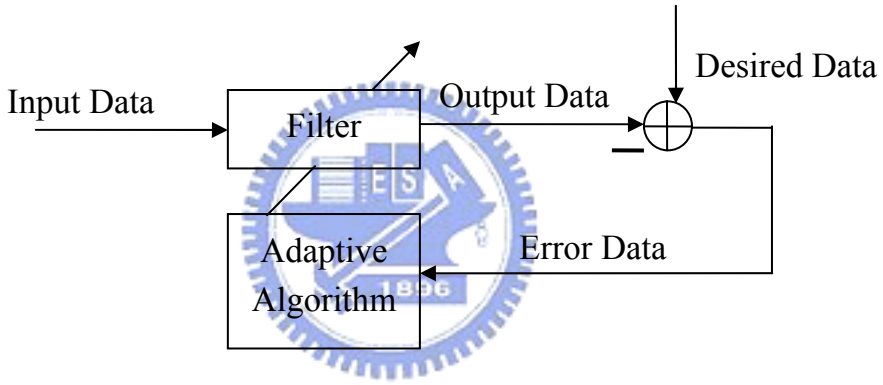

圖 6. 適應訊號架構圖

當訊號輸入適應性濾波器處理之後，輸出訊號與希望達成的訊號不同，產生誤差訊號，將誤差訊號代入適應性演算法，即可調整適應性濾波器的係數，如此經由誤差訊號及適應性演算法不斷的調整適應性濾波器的係數，係數會不斷的變動，最後達到某個穩定的值，此時系統輸出訊號與希望達成的訊號就會非常接近。

2.2.3 LMS Algorithm

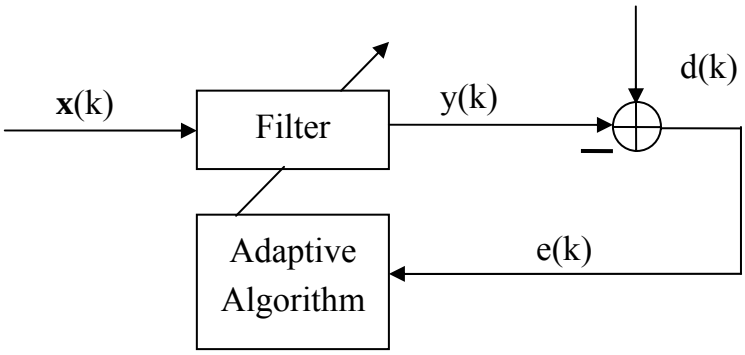

圖 7. LMS 演算法架構圖

$w^H(k)$ 是濾波器在時間 k 的係數。

$$e(k) = d(k) - y(k)$$
$$= d(k) - \mathbf{w}^H(k)\mathbf{x}(k) \qquad (2\text{-}2\text{-}1)$$

$e(k)$ 是 $\mathbf{w}(k)$ 的函數，$-\nabla\mathbf{w}(k)|e(k)|^2$ 的物理意義為：指向 $|e(k)|^2$ 最低處的方向與強度。

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu(-\nabla\mathbf{w}(k)|e(k)|^2) \qquad (2\text{-}2\text{-}2)$$

數學式(3-1-2)的物理意義即為：由 $\mathbf{w}(k)$ 出發，往 $|e(k)|^2$ 最低處的方向前進了 $|\mu(-\nabla\mathbf{w}(k)|e(k)|^2|$ 距離，μ 是一個比重，可以調整前進距離的大小。

$$|e(k)|^2 = e(k)e^*(k) = (d(k) - \mathbf{w}^H(k)\mathbf{x}(k))(d(k) - \mathbf{w}^H(k)\mathbf{x}(k))^*$$
$$= d^2(k) - d(k)\mathbf{w}^T(k)\mathbf{x}^*(k) - d^*(k)\mathbf{w}^H(k)\mathbf{x}(k) + \mathbf{w}^H(k)\mathbf{x}(k)\mathbf{x}^H(k)\mathbf{w}(k)$$

$$\nabla w_r(k)|e(k)|^2 = 0 - d(k)\mathbf{x}^*(k) - d^*(k)\mathbf{x}(k) + \mathbf{x}(k)\mathbf{x}^H(k)\mathbf{w}(k) + \mathbf{x}^*(k)\mathbf{x}^T(k)\mathbf{w}^*(k)$$

$$\nabla w_j(k)|e(k)|^2 = 0 - jd(k)\mathbf{x}^*(k) + jd^*(k)\mathbf{x}(k) - j\mathbf{x}(k)\mathbf{x}^H(k)\mathbf{w}(k)$$
$$+ j\mathbf{x}^*(k)\mathbf{x}^T(k)\mathbf{w}^*(k)$$

$$\nabla \mathbf{w}(k)|e(k)|^2) = \nabla \mathbf{w}_r(k)|e(k)|^2) + \nabla \mathbf{w}_j(k)|e(k)|^2)$$
$$= -2d^*(k)\mathbf{x}(k) + 2\mathbf{x}(k)\mathbf{x}^H(k)\mathbf{w}(k)$$
$$= -2\mathbf{x}(k)(d^*(k) - \mathbf{x}^H(k)\mathbf{w}(k))$$
$$= -2\mathbf{x}(k)e^*(k)$$

所以 LMS Algorithm 即可整理為下列的式子

$$y(k) = \mathbf{w}^H(k)\mathbf{x}(k) \quad \text{=> filter out} \tag{2-2-3}$$

$$e(k) = d(k) - y(k) \quad \text{=> error function} \tag{2-2-4}$$

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu e^*(k)\mathbf{x}(k) \quad \text{=> update weight} \tag{2-2-5}$$

下圖是 LMS Algorithm 的方塊圖：

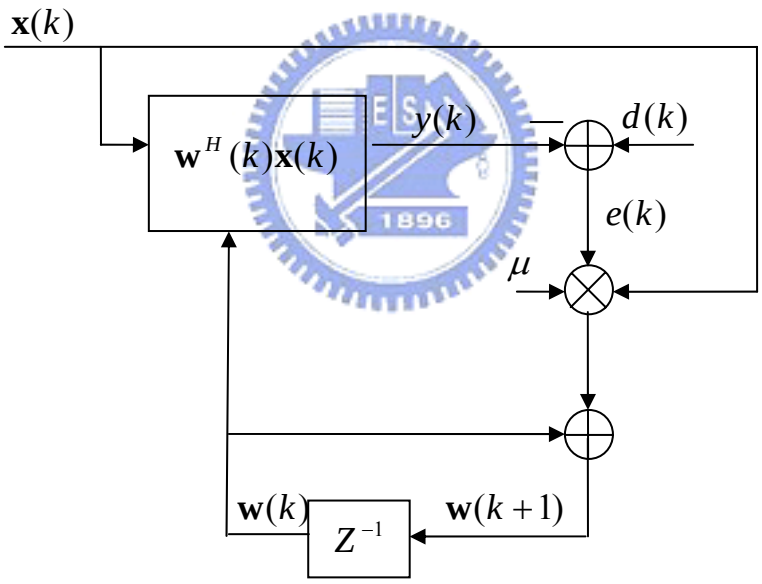

圖 8. LMS 演算法方塊圖

2.2.4 LMS Algorithm 的分析與特性

$$\xi = \sum_k \left|[d(k) - \mathbf{x}^T(k)\mathbf{w}^*]\right|^2 = d_{ss} - \mathbf{p}^H\mathbf{w} - \mathbf{w}^H\mathbf{p} + \mathbf{w}^H\mathbf{R}\mathbf{w} \tag{2-2-6}$$

$$d_{ss} = \sum_k d(k)d^*(k) \qquad \mathbf{R} = \sum_k \mathbf{x}(k)\mathbf{x}^H(k) \qquad \mathbf{p} = \sum_k d(k)\mathbf{x}(k)$$

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu e^*(k)\mathbf{x}(k)$$

$$
\begin{aligned}
E[\mathbf{w}(k+1)] &= E[\mathbf{w}(k)] + \mu E[e^*(k)\mathbf{x}(k)] \\
&= E[\mathbf{w}(k)] + \mu\{\mathbf{R}E[\mathbf{w}(k)] - \mathbf{p}\} \\
&= E[\mathbf{w}(k)] + \frac{1}{2}\mu\nabla E[\mathbf{w}(k)]\xi
\end{aligned} \tag{2-2-7}
$$

(a) LMS 演算法在 $0 < \mu < \frac{2}{\lambda_{\max}}$ 時會收斂，$\lambda_{\max}$ 是 **R** 的最大特徵值。

(b) 如果輸入訊號具有 nonsingular correlation matrix，則 $E[\mathbf{w}(k)] \to w^o$。

(c) 收斂時間為 $\tau \approx \frac{1}{\mu\lambda_{\min}}$

2.2.5 Normalize LMS Algorithm

為了確保 LMS Algorithm 收斂，$0 < \mu < \frac{2}{\lambda_{\max}}$，為了得到 μ 的範圍就必須算出 **R** 的特徵值，並且找出最大的特徵值，如此才能確定收斂範圍。當 **R** 愈大，解特徵值的計算過程就愈複雜。於是有了 Normalize LMS Algorithm 的產生

Normalize LMS Algorithm 如下：

$$y(k) = \mathbf{w}^H(k)\mathbf{x}(k) \quad \text{=> filter out} \tag{2-2-8}$$

$$e(k) = d(k) - y(k) \quad \text{=> error function} \tag{2-2-9}$$

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{\alpha e^*(k)\mathbf{x}(k)}{\gamma + \mathbf{x}^H(k)\mathbf{x}(k)} \quad \text{=> update weight} \tag{2-2-10}$$

只要 $0 < \alpha < 2$，而 γ 是一個微小的常數，其作用只是為了不要讓分母變成 0，如此即可確保演算法收斂，且不必算出算出 **R** 的特徵值。

Normalize LMS Algorithm 能確保收斂的原因在於：

$$Avrg\{\mathbf{x}^H(k)\mathbf{x}(k)\} = \sum_i \lambda_i > \lambda_{\max} \tag{2-2-11}$$

2.3 適應性陣列訊號處理

2.3.1 適應性訊號與陣列式訊號特性比較

陣列式訊號最重要的特性在於「訊號中具備了空間的資訊」，利用這樣的資訊就可以設計波束形成(Beamforming)，達到空間濾波(Spatial Filter)的功能。對於訊號在空間中從不同的角度的輸入，分別給予不同的增益，將不需要的訊號濾除，達到只接收特定方向的訊號。

適應性濾波器具有以下的特性：

適應性濾波器則不需要知道濾波器的特性，只要根據輸出訊號與希望達成的訊號造成的誤差訊號，配合適應訊號演算法，即可改變濾波器特徵，以完成某些特定的需要。而適應性濾波器的特性改變是自動的，並不需要使用者實際干涉調整的過程。

當我們要設計空間濾波器時，必須有以下的假設：

1、 窄頻訊號(Narrow Band)。

2、 遠場平面波(Far Field Plane Wave)。

如此系統輸出能以簡單的數學式表示，許多設計空間濾波器的方法才能使用。

但目前處理的資料是屬於近場球形波的寬頻語音訊號，表示輸出訊號的數學式會變得非常的複雜，而無法使用一般設計空間濾波器的方法，但無論是否滿足假設條件，陣列式訊號必定具備空間上的資訊，空間濾波一定可行，只是一般的設計方法失敗。但適應性濾波器的設計則可以不需要濾波器的特徵，只要根據誤差訊號，配合適應訊號演算法，即可改變適應性濾波器的特性，將適應性濾波器的輸出變成希望達成的訊號。同樣的也可以利用這個方法，針對陣列式訊號處理，設計空間濾波器，達成只接收特定方向的訊號的目的。

2.3.2 設計適應性空間濾波器

設計適應性空間濾波器，首先要知道的是希望輸出訊號的特性為何。我們希望的最後的輸出訊號，是特定方向的語音訊號，而沒有其他方向的訊號，因此首先我們要在安靜的環境之下，經由陣列式麥克風錄下特定方向的語音訊號。

再來是已知的固定干擾源，是希望空間濾波器濾掉的訊號，在桌面環境下一定會存在的干擾源是喇叭的輸出訊號。同樣的也在安靜的環境之下，經由麥克風陣列錄下特定方向的喇叭輸出訊號。

將這兩個訊號存入硬碟中，如下圖所示：

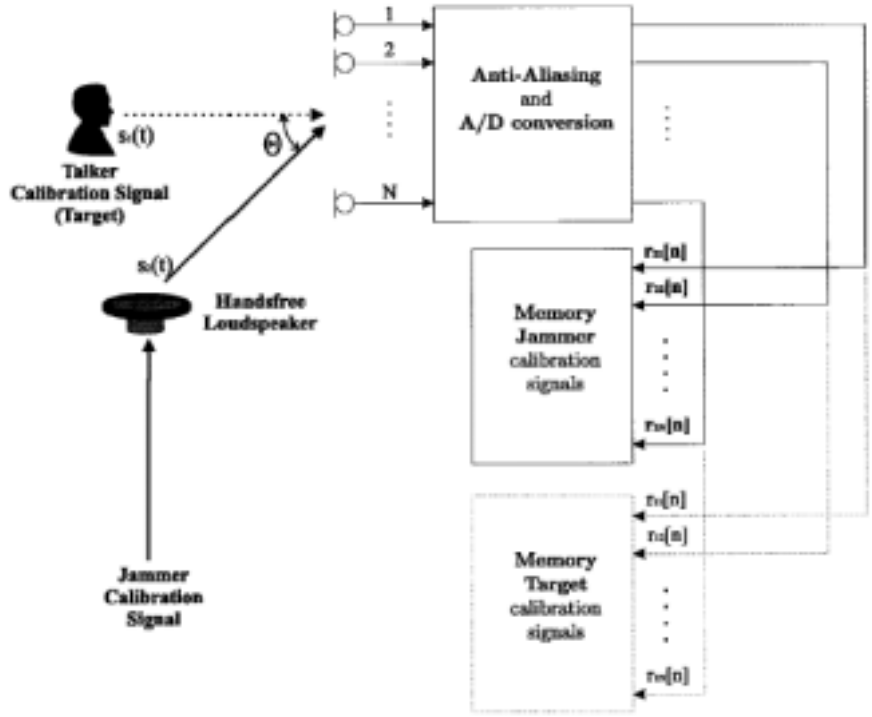

圖 9.錄音儲存方塊圖

適應性陣列式麥克風系統結構如下圖所示：

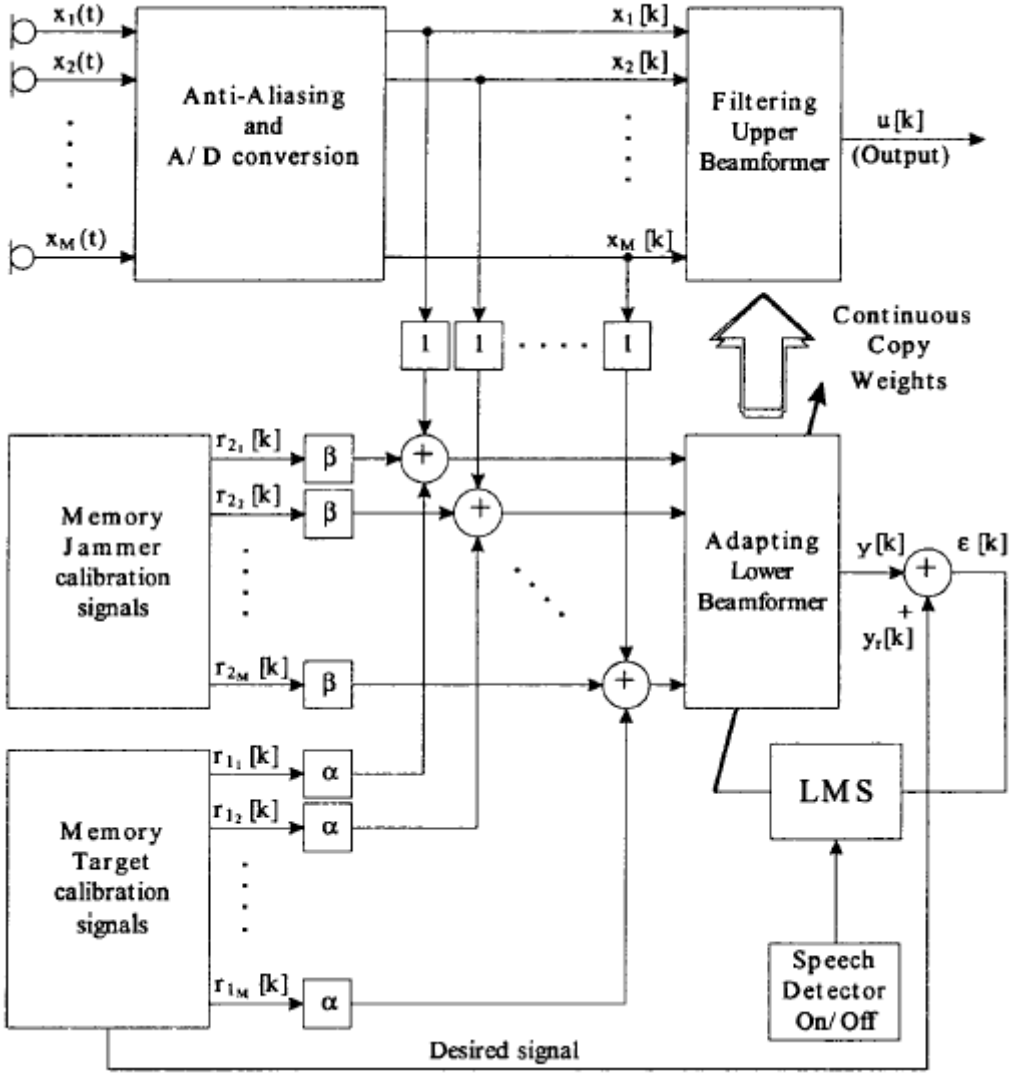

圖 10. 適應性空間波濾器架構圖

將適應性陣列式麥克風系統，移到一般的工作環境中，工作環境中通常具有其他的訊號干擾源，當工作環境中的訊號干擾源不斷輸入，將此訊號與記憶體中的乾淨的語音訊號與乾淨的喇叭訊號各自給與不同的比重相加起來，把這個訊號當成是適應性系統輸入訊號，而在乾淨的語音訊號中選取某一個 channel 的訊號作為希望達到的訊號，利用 Normalize LMS Algorithm 調整適應性系統的係數，係數會不斷的變動，最後收斂到某一個固定的值，如此適應性空間濾波器能夠輸出的訊號與希望達到的訊號之間的誤差最小，即目前的輸出只有預定方向語音訊

號才能夠通過這個適應性空間濾波器，而干擾源方向的訊號通過適應性訊號濾波器時，增益會被壓低，達到濾除干擾源的效果

由使用者判斷是否啟動 Normalize LMS Algorithm，判斷條件為周圍環境是否足夠吵雜，當周圍環境吵雜的程度夠大時，經由 Normalize LMS Algorithm 設計出來的空間濾波器，對於環境中的雜訊干擾源的濾除效果愈好。

因為空間濾波只是針對當時的狀況的訊號干擾源方向加以濾除，濾除方向是固定的，若工作環境中的干擾源方向移動，則必須重新再啟動 Normalize LMS Algorithm，更新適應性空間濾波器的係數，重新設計要濾除干擾源的方向，如此即可再次濾除干擾源。

第三章 實驗平臺

3.1 實驗平臺架構

實驗平臺架構主要可分為四大部分：

1、 麥克風前端濾波及放大電路

2、 資料拮取電路

3、 USB 傳輸裝置

4、 Host 端 Driver 及 Software

如下圖：

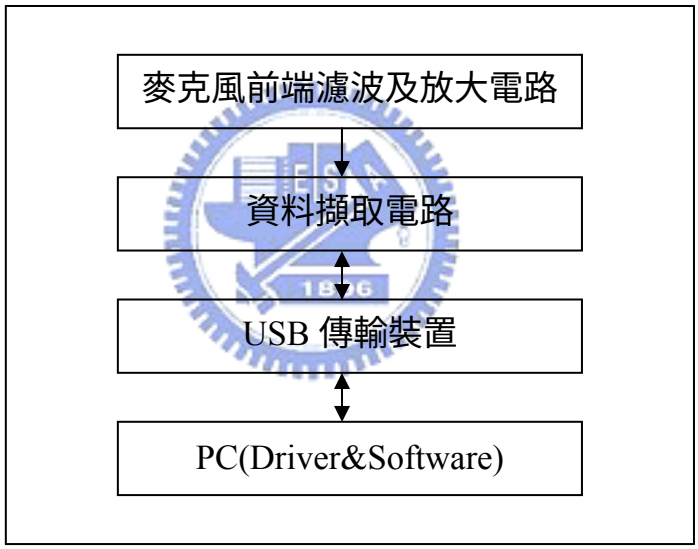

圖 11. 實驗平台架構圖

3.2 麥克風前端濾波及放大電路：

方塊圖：

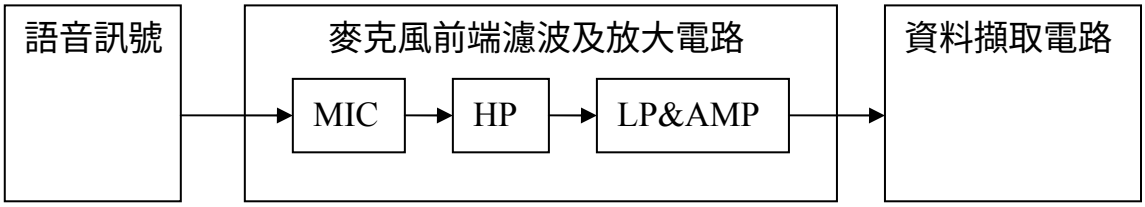

圖 12. 麥克風前端濾波器及放大電路架構圖

電容式麥克風將語音訊號轉換為電壓訊號後，必須先通過 High-Pass Filter，主要原因有二：

1、 麥克風輸出的訊號除了語音訊號之外，另外帶有 5V 的直流訊號。為了避免放大器將直流訊號也一起放大，必須先通過 High-Pass Filter，濾除直流電壓部分。

2、 麥克風對於低頻的聲音相當的敏感，如空氣的擾動，而這樣的雜訊也往往使得語音訊號失真。

在此，我們將 High-Pass Filter 的 3dB 點定在 40Hz，濾除掉大部分的低頻雜訊，並保留住大部分的語音資訊。

訊號通過 High-Pass Filter 之後，訊號非常小，peck-to-peck 電壓值只有幾十個 mV，因此加上了放大器，將訊號放大到-2.5V~2.5V 之間。語音訊號在後段 A/D 的 Sampling Rate 為 16KHz，根據 Nyquest Sampling Rate，Sampling Rate 至少要是訊號頻帶的兩倍，訊號才不會有 Aliasing 的問題，因此要再加上一個 Low-Pass Filter 將 8KHz 以上的訊號濾除。

經過濾波與放大電路後，在安靜的實驗室的環境下的輸出訊號 peck-to-peck 值約 50mV。

放大器及濾波器電路圖：

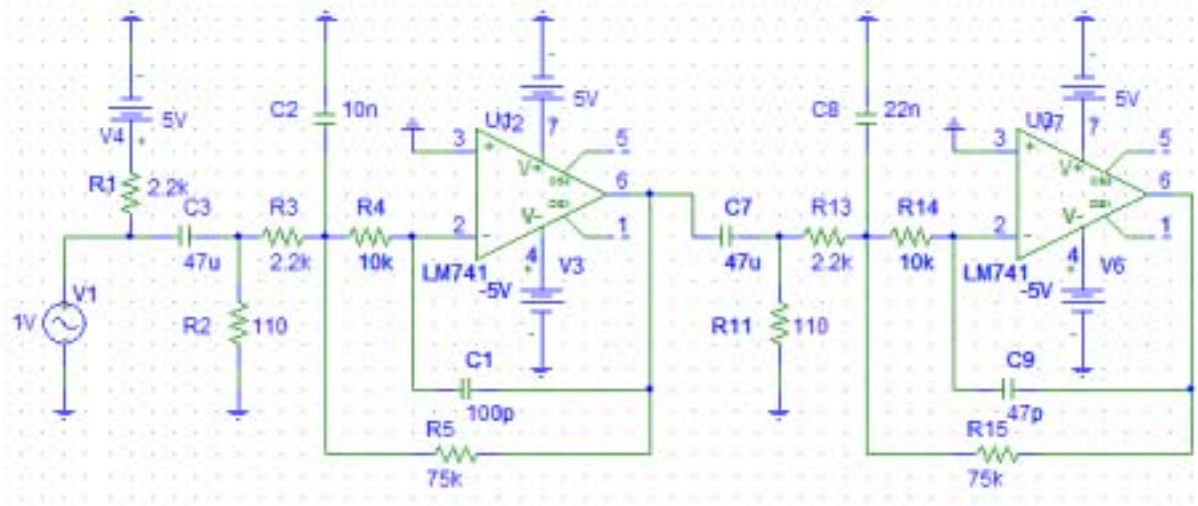

圖 13. 放大器及濾波器電路圖

模擬電路的頻率響應圖：

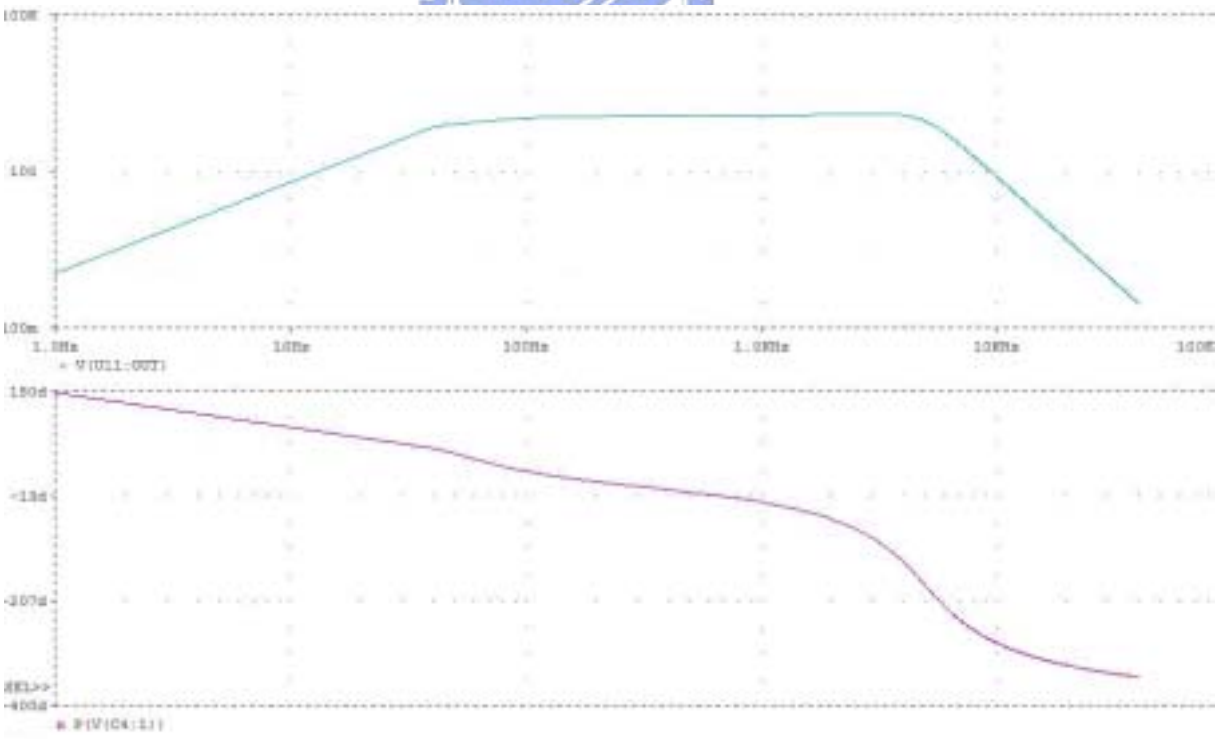

圖 14. 放大器及濾波器模擬電路的頻率響應圖

3.3 資料擷取電路

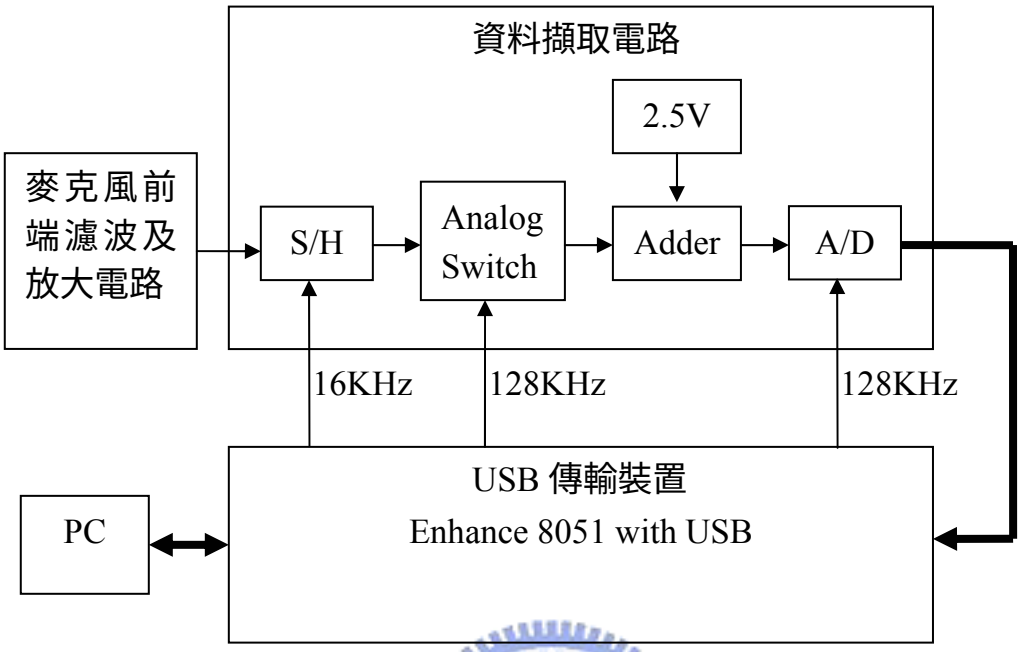

圖 15. 資料擷取電路架構圖

S/H：Sample and Hold，型號為 LF398AN。

輸入訊號：麥克風通過濾波及放大電路的訊號。

控制訊號：8051 的 IO。

輸出訊號：當控制訊號為 Low，輸出訊號為控制訊號由 High 變 Low 瞬間的輸入訊號電壓。

訊號輸入裝置為陣列式麥克風，陣列式麥克風訊號最重要的資訊來自於：不同方向的聲音訊號輸入，因為相對位置的不同，而對於每個麥克風造成的 Delay 不同。A/D 將類比訊號轉換為數位元訊號時需要時間，對於每個 channel 做 A/D

的順序的不同，會造成不同的額外的 Delay。因此需要 S/H 這樣的裝置，在同一個時間點，維持住所有的麥克風輸入的電壓(Snapshot)，如此才能完整的保留住輸入訊號對於不同的麥克風造成的 Delay。

8051 給 S/H 的控制訊號以 16KHz 的速度切換，即每次 Snapshot 的 Sampling Rate 為 16KHz。

Analog Switch：型號為 CD74HC4051E。

輸入訊號：8 個 Channel 的 S/H 的輸出訊號。

控制訊號：控制訊號：8051 的 IO。

輸出訊號：由 8051 控制的 3bits 訊號，可以切換 8 個不同的訊號來源做為輸出。

Analog Switch 本身在控制切換的瞬間，在輸出訊號上產生突波，突波的能量雖然不大，但還是會影響到最後的輸出，造成輸出雜訊變大。

輸入 channel 共 8 組，因此 8051 給 Analog Switch 的控制訊號，切換頻率為 128KHz。

Adder：將輸入訊號由-2.5V~2.5V 提升到 0V~5V，是為了配合 A/D 的 Sample 範圍。

A/D：Analog to Digital Converter。型號：ADS8323。

Maximum Sampling Rate：500KHz。

每筆資料儲存位元：16bits。

Sample Range：0V~5V。

抓進來的資料存到 ISO Buffer 時先存 LSB 再存 MSB，如此主機接收資料後只要用 16bits 有號數指標讀取，即可讀到傳回的值。

控制時序圖如下：

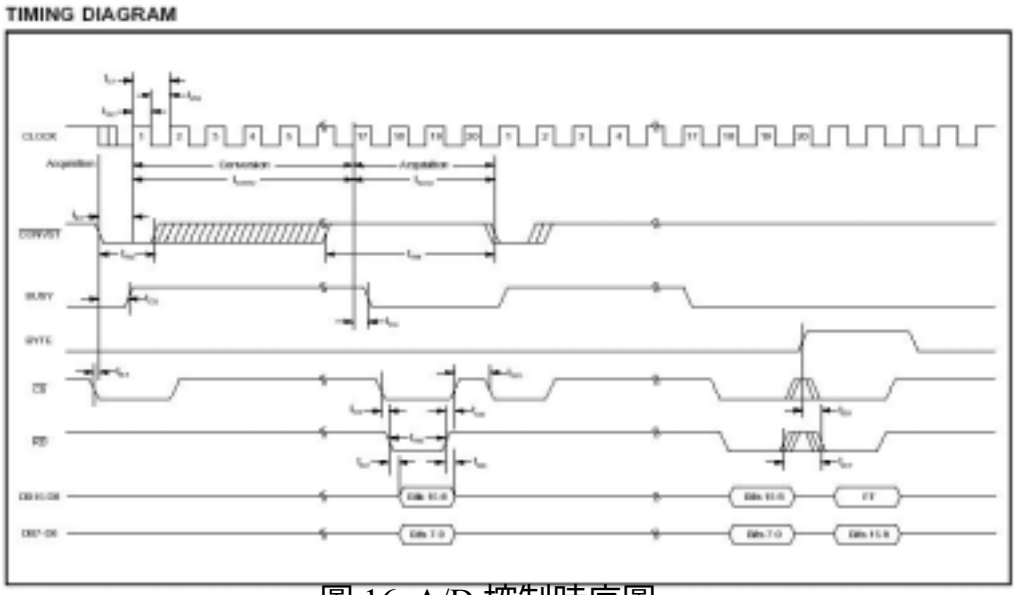

圖 16. A/D 控制時序圖

3.4 USB 傳輸裝置

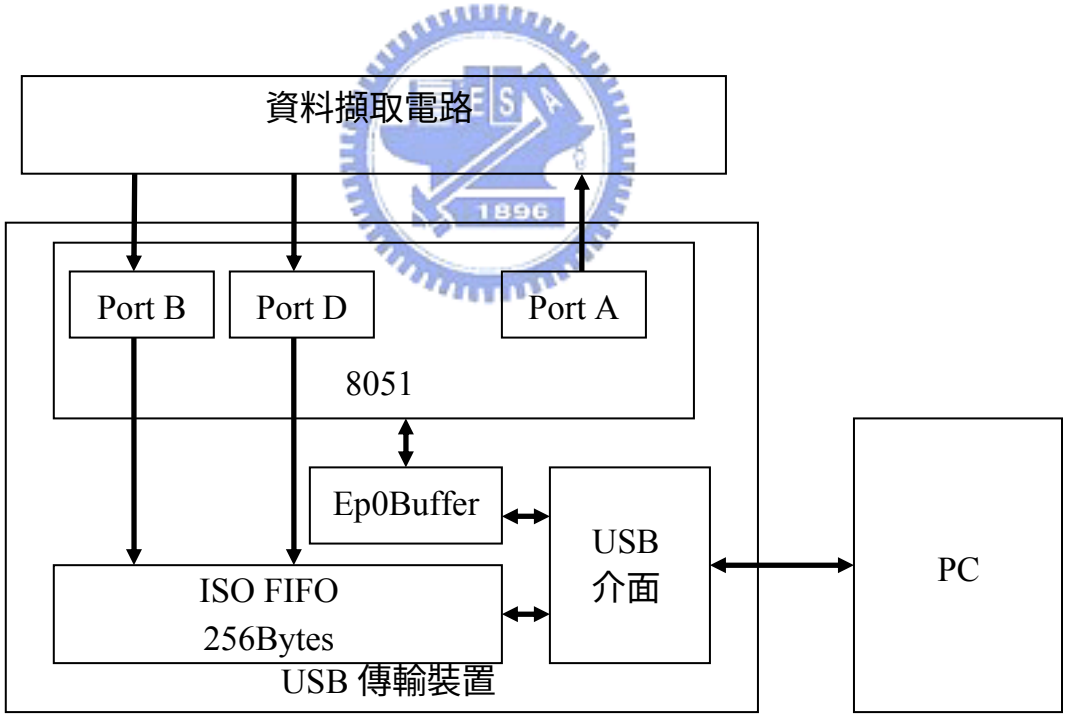

圖 17. USB 傳輸架構圖

在平臺架構中，使用 Cypress EZ-USB FX，具有 USB 介面的增強型 8051，作為 USB 傳輸裝置，將資料擷取電路得到的數位訊號，經由 USB 介面傳給主機。

3.4.1 USB 通訊協定

USB 的通訊協定主要可以分為三種封包(Package)：

1、執照封包(Token)：指示型態與方向。包含了 PID、Address、Endpoint、CRC。PID 包含 IN、OUT、Setup、SOF。位址跟端點提供接收裝置的位址及接收的端點。CRC 檢測此封包是否接收正確。無論何種傳輸，每次傳輸的第一個封包一定為執照封包。

(1)、IN：裝置送封包給主機。

(2)、OUT：主機送封包給裝置。

(3)、Setup：僅給控制傳輸使用，用來確認裝置要求。

(4)、SOF：每 1mS 發生一次，用來表示 USB Frame 的抵達。

2、資料封包(Data)：實際上的傳輸資料。包含了 PID、Data、CRC。PID 包含 Data0、Data1。Data 則為實際上的傳輸資料。CRC 檢測此封包是否接收正確。

(1)、Data0，Data1，兩種封包會交替傳送，透過主機內部的 Toggle 位元狀態與 DATA PID 相比較，可偵測到錯誤的交握封包。

3、交握封包(Hand-Shake)：USB 的傳輸狀態，有四種 PID，除了等時傳輸之外，每次傳輸的最後一個封包必為交握封包。

(1)、ACK：成功。正確無誤的接收資料。

(2)、NAK：忙線。再試一次。

(3)、STALL：停滯。發生無法預料的問題，表示不瞭解裝置的需求。

(4)、NYET：端點忙線。資料傳輸成功，但端點尚未準備好接收下一筆資料。

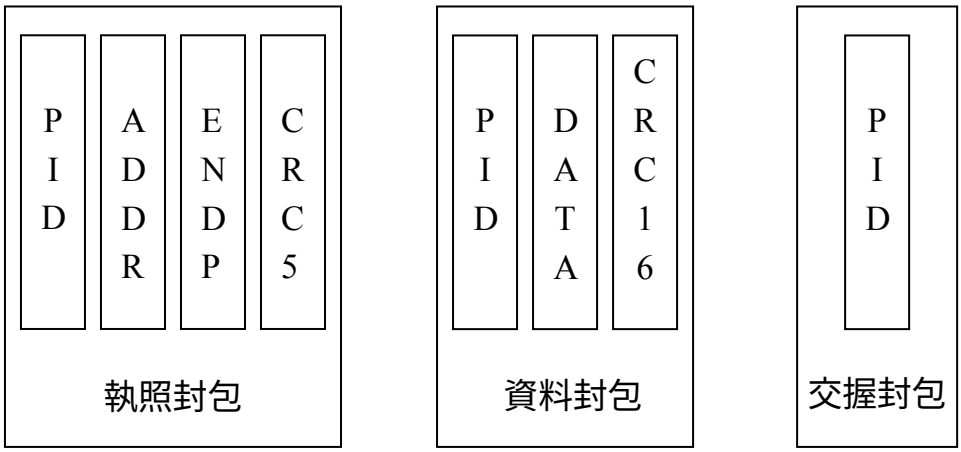

圖 18. USB 封包類型

封包的接收及發送由 USB 介面處理。

3.4.2 USB 傳輸模式

USB 系統提供了四種不同的傳輸模式：

1、 巨量傳輸(Bulk)：突發性的傳輸模式。資料封包大小為 8、16、32、64 位元組。除了資料封包之外另有交握封包(Hand-Shake Package)，及自動錯誤資料檢核機制(CRC)，如資料傳送錯誤，可要求裝置重送封包，確保資料的正確性。

2、 中斷傳輸(Interrupt)：類似巨量傳輸，資料封包大小為 1~64 位元組。高速的裝置中。需經由主機規則固定間隔詢問。

3、 等時傳輸(Isochronous)：在固定的時間傳出封包，主要使用在音頻與影像等資料流中。為了確保封包可以在固定的時間送出，無 Hand-Shaking 封包，僅具有 CRC 錯誤檢核，資料傳輸錯誤亦不再重送封包。時間是最重要的要求條件。

4、 控制傳輸(Control)：用來配置及傳送命令給裝置，確認裝置要求。

而在實驗平臺的架設中，只用到控制傳輸及等時傳輸。

	封包大小(Byte)	時間	資料檢查	應用
巨量傳輸	8、16、32、64	盡快完成	有	儲存裝置、印表機
中斷傳輸	1　64	1　255ms	有	滑鼠、鍵盤
等時傳輸	1　1024	1ms	沒有	影像、聲音
控制傳輸		盡快完成	有	命令

表 1. USB 四種傳輸模式比較

3.4.3 USB 裝置韌體

USB 裝置韌體流程圖：

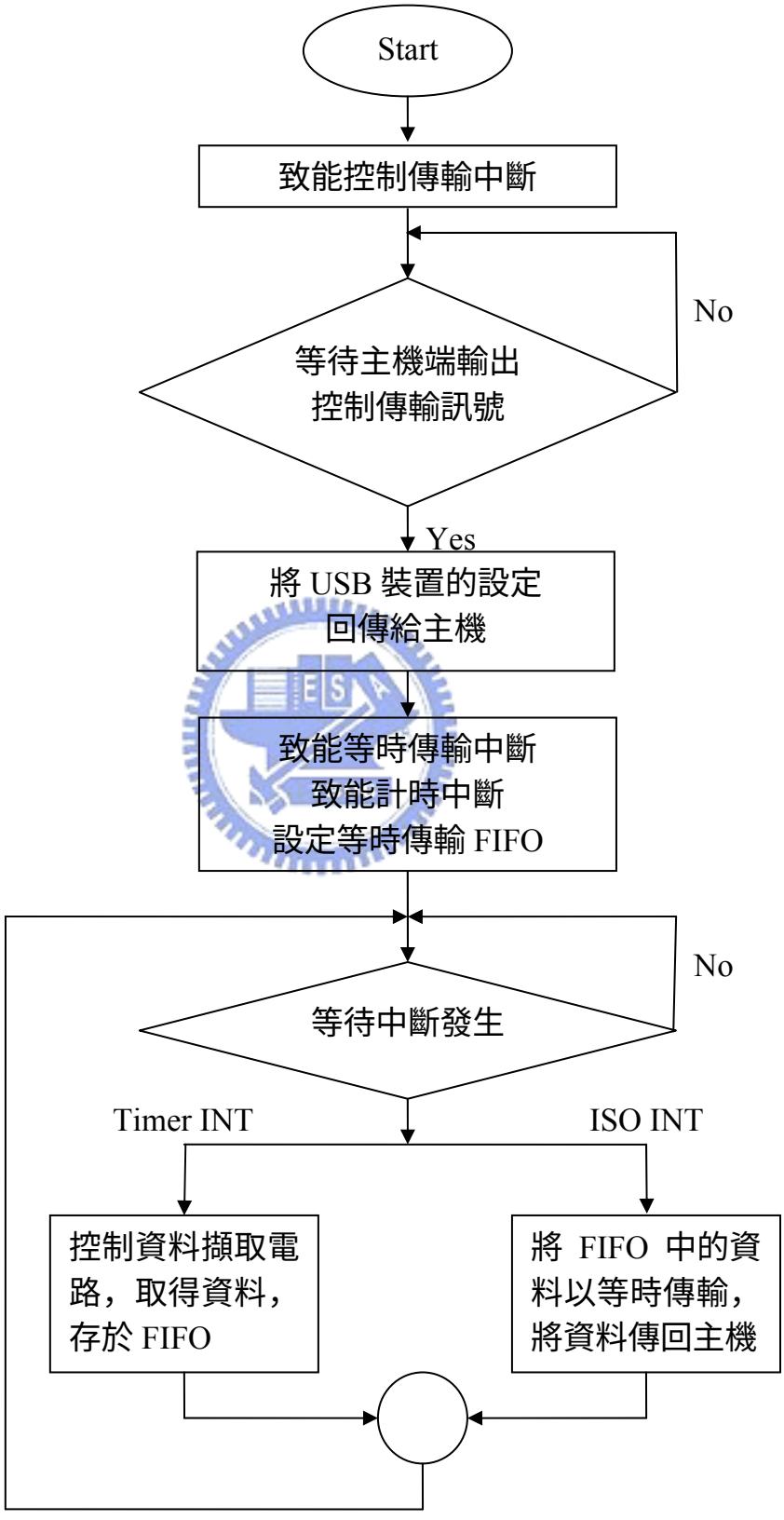

圖 19. USB 裝置韌體流程圖

當韌體燒入 USB 裝置的 RAM 之後，以 USB 接線與主機連接。主機經由控制傳輸要求 USB 裝置回傳裝置描述元，主機由裝置描述元的資訊，確認 USB 裝置，並使用相對應的驅動程式來驅動 USB 裝置，回傳的裝置描述元主要有下列幾項：

1、 Device Descriptor：規格版本、裝置群組、VID、PID、配置數目。

2、 Configuration Descriptor：端點長度、界面數目、界面數值、電源屬性、所需電源。

3、 Interface Descriptors：切換設定、端點數目、界面群組。

4、 Endpoint Descriptors：使用端點及方向、端點型態、最大封包大小、輪詢間隔。

5、 String Descriptors：公司、產品名稱。

在實驗平臺架構中，我們使用 Cypress Sample Device Driver 驅動 USB 裝置，根據 Cypress Sample Device Driver 所需要的裝置描述元設定，由控制傳輸回傳裝置描述元，即可成功讓主機確認 USB 裝置，使用相對應的驅動程式。

主機成功確認 USB 裝置之後，啟動等時傳輸中斷及計時中斷，計時中斷的產生頻率為 16KHz，等時傳輸中斷產生的頻率為 1KHz。

當計時中斷發生時，8051 IO 控制 8 個 S/H 在同一個時間，將 8 個 channel 的輸入訊號維持住，接著控制 Analog Switch 切換不同 channel 的輸入訊號，輸出控制訊號給 A/D，將類比訊號轉為數位訊號，最後將 A/D 產生的 16bits 的數位訊號，由 IO 讀入 8051，再存入 ISO FIFO(先存 LSB 再存 MSB)，如此重覆 8 次，即可將同一個時間的 8 個 channel 的值由類比訊號轉為數位訊號，存入 ISO FIFO，等待等時傳輸中斷發生。

當等時傳輸中斷發生時，USB 介面會將 ISO FIFO 中的資料，以等時傳輸的方式傳回主機。韌體校正計時器的值，使計時中斷與等時傳輸中斷同步。8051 的 IO 控制資料擷取電路，將 8 個 channel 的資料取回，再存入 ISO FIFO。

每次完整的等時傳輸資料，由 1 次等時傳輸中斷及 15 次計時中斷，將 8 個 channel 的語音訊號，經由 8051 IO 控制資料擷取電路，將每筆 2Bytes 的資料存入 ISO FIFO，因此等時傳輸的資料長度為 256Bytes, 等時傳輸的傳輸速率為 256KB/S。

計時中斷與等時傳輸中斷同步：

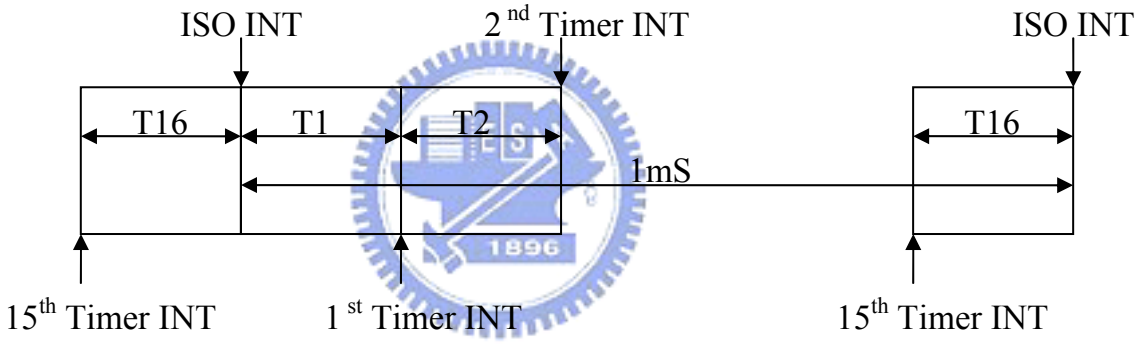

圖 20. 計時中斷與等時傳輸中斷同步說明

T2~T15 的時間為計時中斷發生間隔，略大於 1/16mS，而 T1、T16 則平分了剩下的(1mS-T2-…-T15)的時間，T1 由等時傳輸中斷副程式校正，如此才能確保兩個中斷同步。

3.5 Driver & Software

3.5.1 Driver

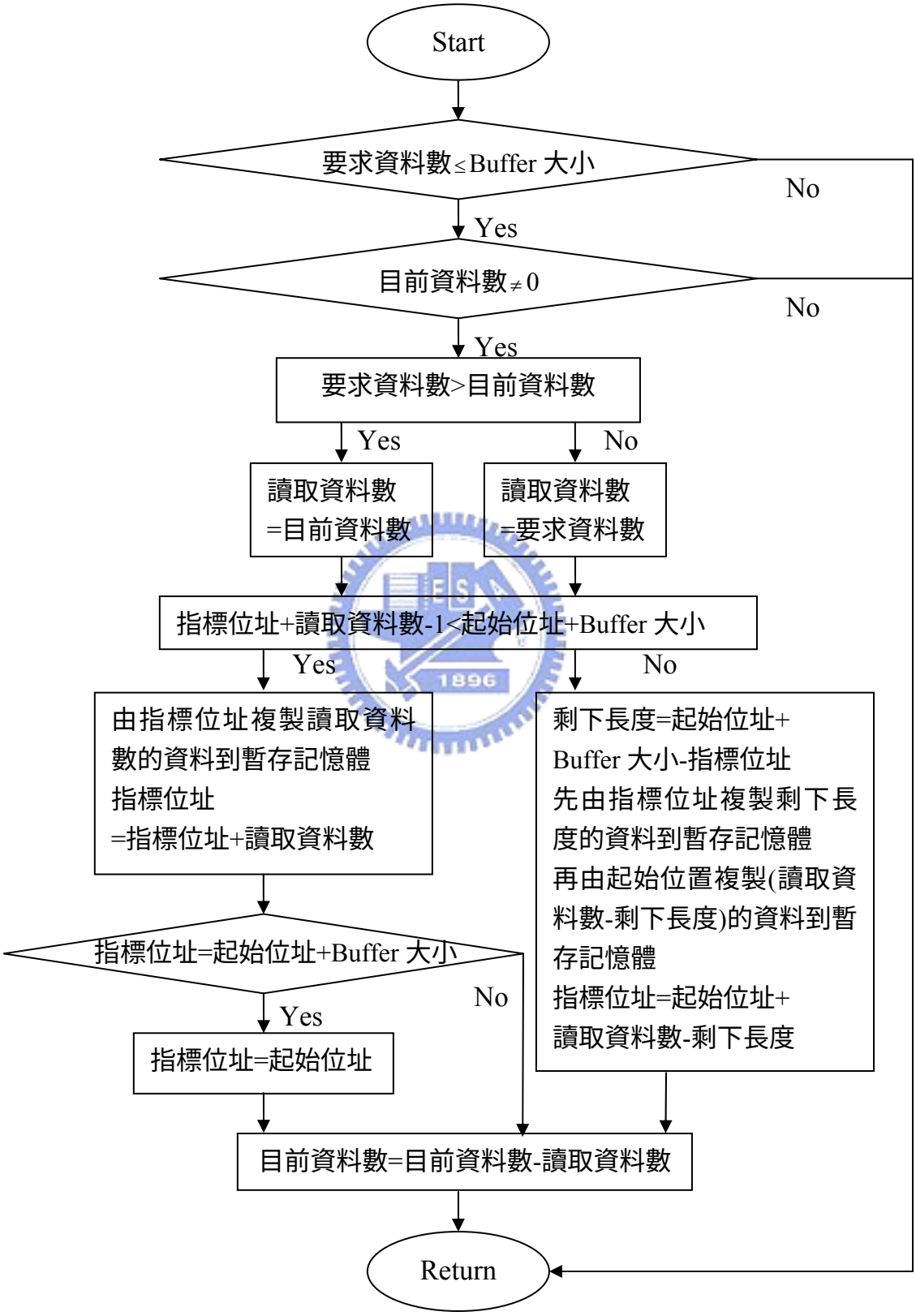

圖 21. Ring Buffer 接收 USB 裝置的資料流程圖

3.5.2 軟體流程圖：

軟體主要以 Switch Case，不斷掃描按鍵是否輸入，搭配 Thread 來重覆執行想做的命令，如要停止 Thread 則按下 Stop Thread 即可結束動作。

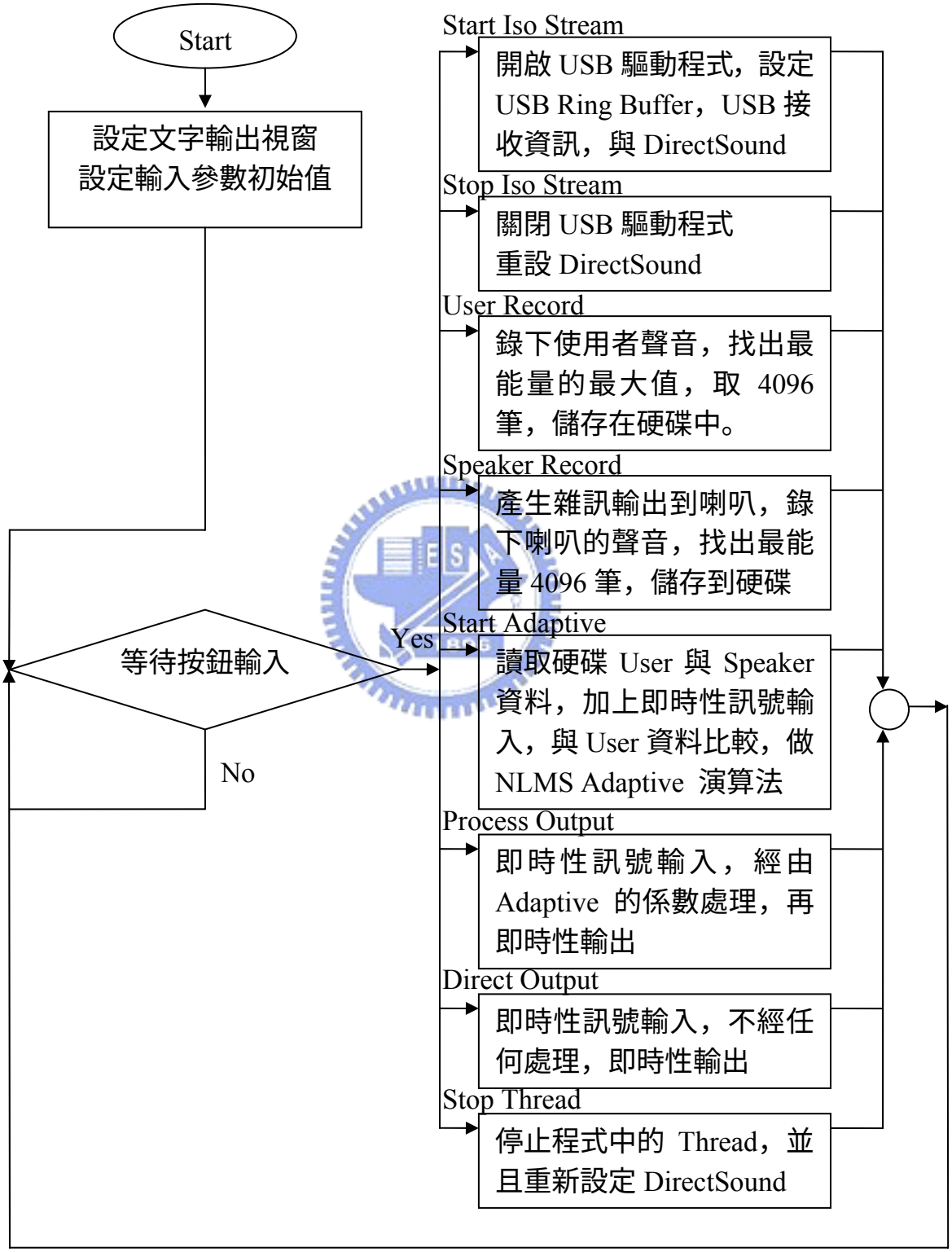

圖 22. 軟體流程圖

3.5.3 使用者介面

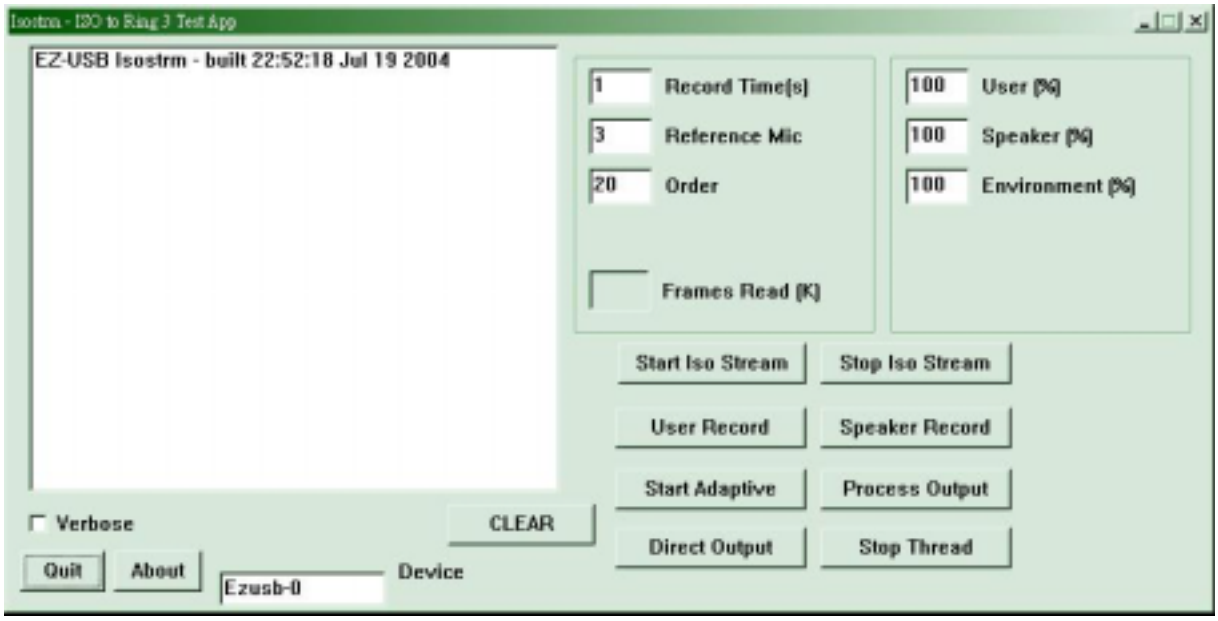

圖 23. 軟體使用者介面

1、 Start Iso Stream：開啟驅動程式，設定驅動程式 Ring Buffer 的記憶體大小、設定 USB 接收資訊、設定 DirectSound。

2、 Stop Iso Stream：關閉驅動程式，重設 DirectSound。

3、 User Record：利用 USB 裝置及相關電路，錄下使用者聲音，找出最能量的最大值，取 4096 筆，儲存在硬碟中。

4、 Speaker Record：程式產生雜訊，輸出到喇叭，再利用 USB 裝置及相關電路，錄下喇叭的聲音，找出最能量的最大值，取 4096 筆，儲存在硬碟中。

5、 Start Adaptive：將硬碟中使用者與喇叭的資料，分別讀取至 User Buffer 與 Speaker Buffer。再實際上的輸入，加上 User Buffer 與 Speaker Buffer 的資料，以 NLMS Adaptive 的方式，經由 4096 次的 iteration 之後，得到最後的係數。

6、 Process Output：將輸入的訊號，經過 NLMS Adaptive 算出的係數處理，最後再由喇叭輸出。

7、 Direct Output：語音訊號利用 USB 裝置及相關電路輸入主機之後，不做任何處理，直接由喇叭輸出。

8、 Stop Thread：停止程式中的 Thread，並且重新設定 DirectSound。

9、 Record Time：要錄下多少秒的聲音。

10、 Reference Mic：做 Normalized LMS Adaptive Algorithm 的時候，會以那一個麥克風輸入做為比較。

11、 Frames Read：目前收到多少 Frames，顯示於介面上。

12、 User、Speaker、Environment：做 Normalized LMS Adaptive Algorithm 的時候，可以調整比重，讓 Adaptive 針對需要的功能加強效果。

13、 Device：裝置名稱，用來開啟 USB 裝置時需要的名稱。

14、 Clean：清除訊息框的資訊。

15、 Quit：關閉使用者介面。

3.5.4 使用流程

1、 按下 Start Stream：開啟驅動程式，設定驅動程式 Ring Buffer 的記憶體大小、設定 USB 接收資訊、設定 DirectSound。如此驅動程式才能正常的接收 USB 裝置傳回的資料。設定 DirectSound 則可以使喇叭持續的發出 DirectSound Ring Buffer 中的資料。

2、 在安靜的環境之下，使用者發出聲音，同時按下 User Record，如此陣列式麥克風即可錄下使用者發出的聲音。由錄下的聲音中尋找能量最大附近的 4096 筆資料，再將這 4096 筆資料儲存到硬碟。

3、 在安靜的環境之下，按下 Speaker Record，程式會產生 Uniforn 雜訊，並使用 Thread 使喇叭不斷發出雜訊，如此陣列式麥克風即可錄下喇叭發出的聲音。

4、 到吵雜的環境之中，按下 Start Adaptive，即可輸入吵雜環境的聲音，並與之前在乾淨的環境中錄下的使用者與喇叭的資料，以 Normalize LMS Adaptive Alogrithm，經過 4096 次的 Iteration 後，即可得到最後的係數。

5、 按下 Process Output，吵雜的環境之中，使用者在錄取聲音的同樣的角度說話，即可發現此時由喇叭輸出的聲音，吵雜環境的聲音被抑制了，而喇叭發出的聲音也不會經由麥克風再次輸入主機，而使用者的聲音是清楚乾淨的。

6、 按下 Direct Output，吵雜環境、喇叭、使用者的聲音，不斷經由麥克風輸入主機，並由喇叭持續不斷輸出，明顯的由喇叭發出的聲音中包含了所有未經處理的資訊。可與步驟 5 比較其輸出結果。

3.5.5 DirectSound

目前視窗多媒體程式，大多運用 DirectX 來設計，主要原因是因為透過 DirectX 的使用，程式可以直接與電腦本身的硬體溝通，以提升程式效能。

DirectSound 是 DirectX 中，提供 Wave 類型的音效處理介面，音效處理的觀念如下：

1、 裝置物件(Device Object)：DirectSound 裝置物件是在程式中呼叫 DirectX API 的函數來建立，它是一個代表程式所使用的音效輸出入裝置的物件，在物件裝置之後才能進一步設定裝置的協調層級，建立主緩衝區與次緩衝區。

2、 協調層級(Cooperative Level)：Windows 是一個多工作業系統，在同一時間可能會多程式用到相同的硬體資源，因此當音效裝置物件建立之後，還要設定程式對於裝置的使用權限，也就是協調層級。

3、 主緩衝區(Primary Buffer)：主緩衝區是用來儲存要播放聲音資料的記憶體區域。當裝置物件建立時，主緩衝區便會自動產生。

4、 次緩衝區(Secondary Buffer)：次緩衝區是儲存聲音資料記憶體區域，可以有很多個，其中的聲音資料可被放置到主緩衝區來播放。依播放格式設定次緩衝區的主要參數，如取樣頻率、每筆資料大小、一秒鐘的資料量。

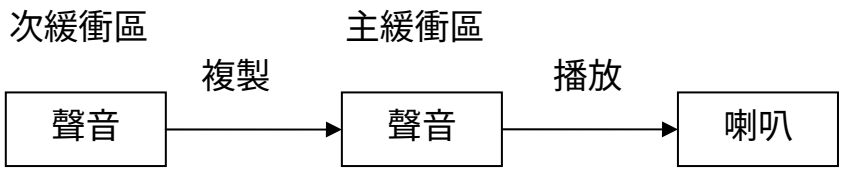

圖 24. DirectX-DirectSound 架構圖

載入資料到次緩衝區：

要將聲音載入到次緩衝區中，必須進行三個動作：鎖定次緩衝區、載入資料到次緩衝區、解除鎖定次緩區。在此過程中鎖定次緩衝區的用意，是為了防止資料尚未複製到次緩衝區之前的任何不當的存取動作。

靜態緩衝區與串流緩衝區：

靜態緩衝區與串流緩衝區均屬於次緩衝區，靜態緩衝區播放的聲音是短促的、預定的聲音；如要播放連續不間斷的、非預定的聲音，則需設定串流緩衝區。在實驗平臺的聲音輸入中，聲音輸出長度未定，輸入時間未定，因此將次緩衝區宣告為串流緩衝區。靜態緩衝區與串流緩衝區設定不同之處在於 dwFlags 旗標，靜態緩衝區的 dwFlags 旗標設定為 DSBCAPS_STATIC；串流緩衝區的 dwFlags 旗標設定為 DSBCAPS_GETCURRENTPOSTION2。

此外也也希望當程式切換到其他程式時，仍可繼續播放聲音，因此 dwFlags 旗標也須開啟 DSBCAPS_GLOBALFOCUS。

緩衝區的播放設定要啟動 Loop 的功能。串流緩衝區複製資料到主緩衝區時是用 Ring Buffer 的方式，當資料複製已經達到串流緩衝區的底部時，會馬上由串流緩衝區的起始點，重新複製新的資料給主緩衝區。

當播放命令啟動後，聲音資料不斷的由次緩衝區複製到主緩衝區，需要同步進行資料的傳遞與緩衝區中現行播放位置的通知，如此載入資料到次緩衝區後，即可馬上被複製到主緩衝區，將聲音播放出來，而不會有無法預測的 Delay。要得知緩衝區中現在播放位置，有兩種方法可以用：定期詢問播放位置(Polling)以及通知(Notification)。前者是主動的不斷的詢問目前位置，後者則是當播放位置達到我們感興趣的位置，以訊號事件通知。

每次載入次緩衝區的資料固定為 256Bytes 的倍數，宣告的次緩衝區大小宣告成 256 的倍數為 4096Bytes，資料輸入主機的速度與播放速率相同，均為 16KHz。所以只要確定第一次寫入時的播放位置位於何處，之後將資料不斷的寫入，使用次緩衝區的方式如同使用 Ring Buffer, 當寫入的累計資料長度大於次緩衝區的大小，將指標移到次緩衝區的起始處，把剩下的超過的資料長度再寫入，如此不斷的重覆寫入資料即可。

第四章 實驗結果與分析

在桌面環境下，測試麥克風數目、雜訊源、與 Adaptive FIR 階數的關係

使用者：與麥克風陣列中心距離 30 公分，夾角 90 度
雜訊源 1：與麥克風陣列中心距離 37 公分，夾角 30 度
麥克風陣列距離桌面 27 公分 麥克風陣列距離反射面 60 公分
如下圖所示：

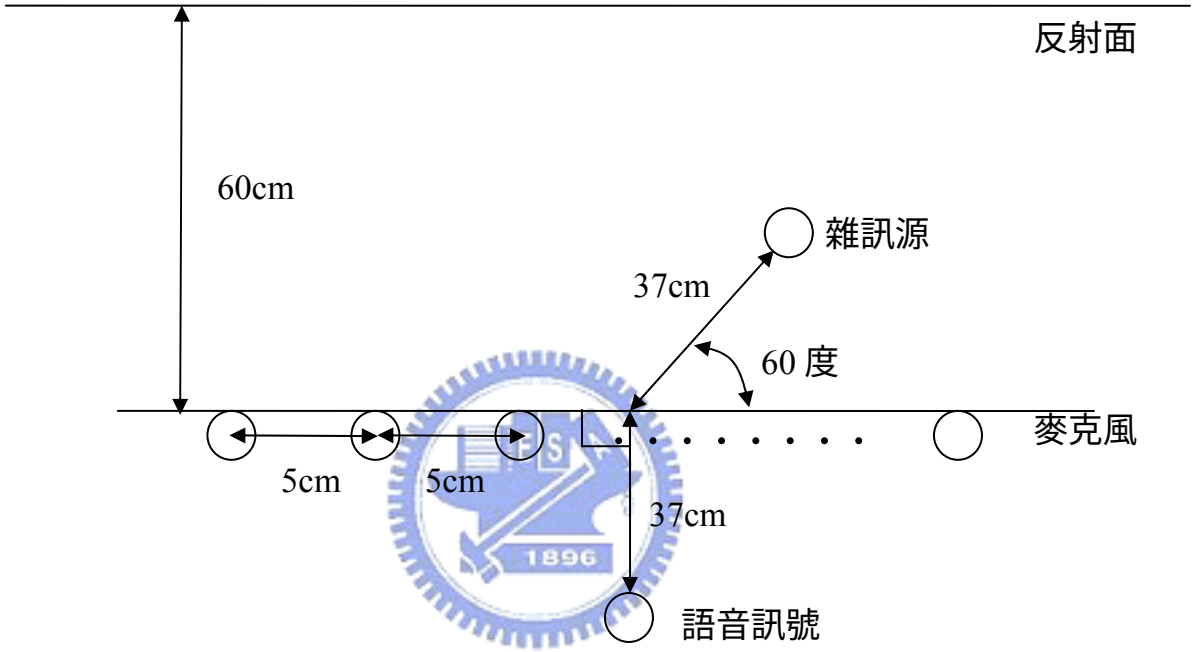

圖 25. 實驗環境說明-干擾源個數：1

計算訊號 SNR 的方法如下：

$$\text{平均訊號能量}=\frac{\sum_{i=M}^{N} x^2(i)}{N-M+1}$$

在實驗中，我們將單純的雜訊的部分:第 7001 筆到第 10000 筆的資料，利用上式算出雜訊在某一段區間的平均能量。同理，在第 29051 筆到第 32500 筆的資料為語音訊號與雜訊與雜訊的部分，同樣的算出語音訊號與雜訊的平均能量，兩者相減即可得到訊號的 SNR。

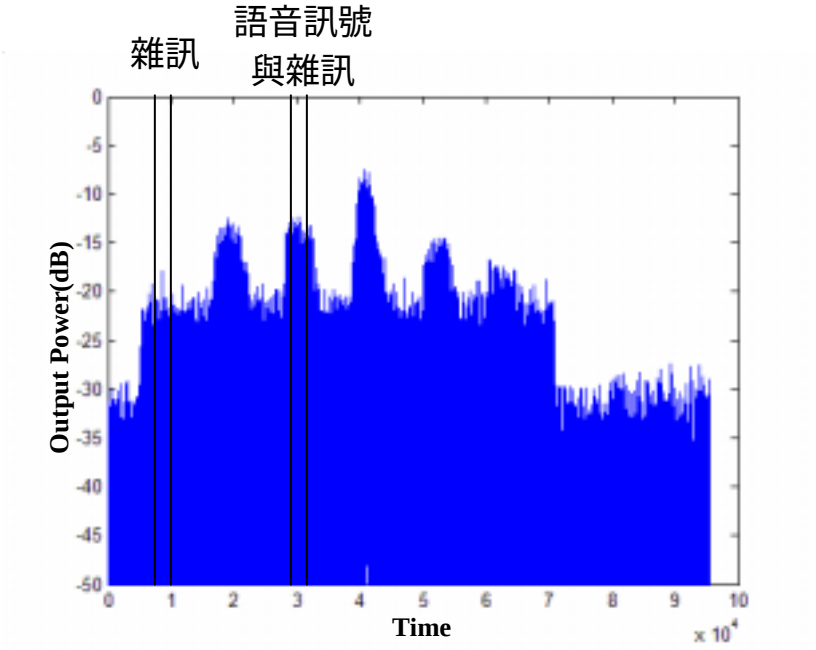

圖 26. 雜訊與語音訊號平均能量的區間

未經處理的訊號如下：

實驗室環境下，一個雜訊源干擾時：

語音訊號與雜訊：-18.8012dB

雜訊：-30.2613dB

SNR=11.4601dB

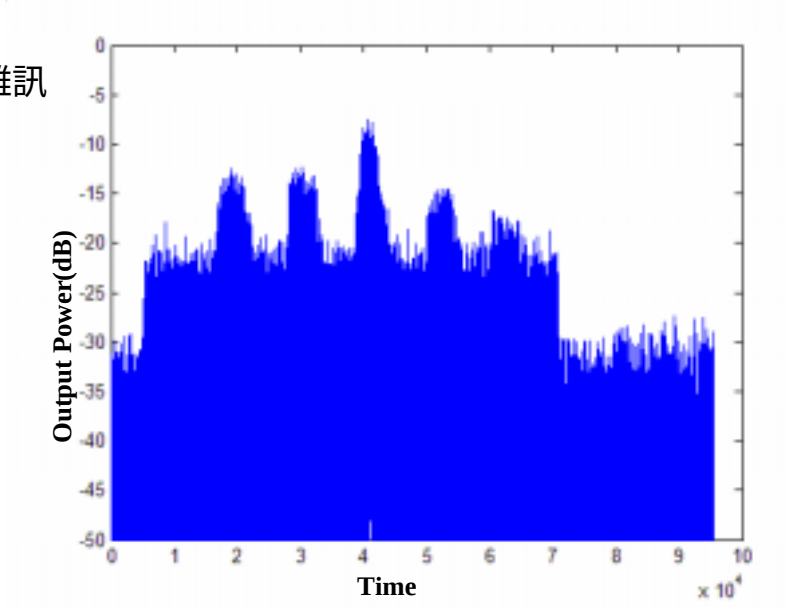

圖 27. 未經處理的訊號-干擾源個數：1

4.1 實驗一、比較 FIR 階數與 SNR 的關係

使用 NLSM Algorithm
陣列式麥克風個數：8
FIR 階數：10

處理後的資料
SNR=19.5390dB

SNR 比處理前增加
8.0790dB

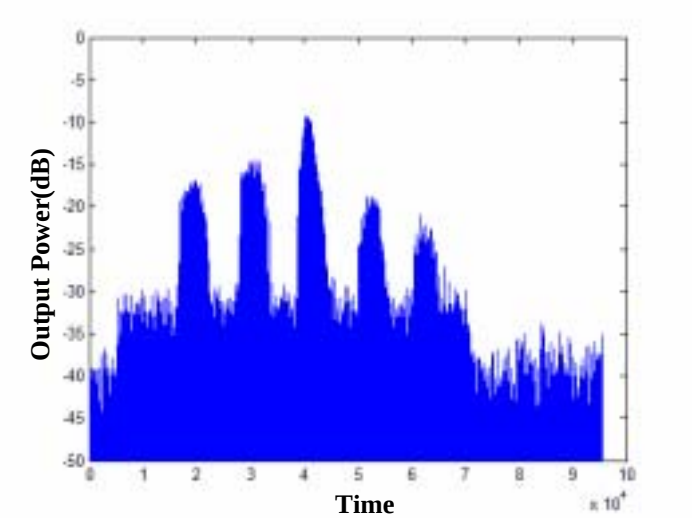

圖 28. 適應空間濾器輸出-麥克風個數 8、階數 10

係數：

0.0624	0.0510	0.0811	0.0414	0.0772	0.0520	0.0083	-0.0730
0.0197	0.0389	0.0393	-0.0018	0.0330	0.0475	0.0503	-0.0007
0.0072	0.0235	0.0122	-0.0473	0.0149	-0.0024	0.0170	0.0450
-0.0065	-0.0080	-0.0137	-0.0202	0.0102	-0.0247	0.0059	0.0465
0.0081	-0.0027	0.0153	0.0051	0.0161	-0.0039	0.0254	0.0315
0.0316	0.0017	0.0320	-0.0037	0.0171	0.0001	0.0400	0.0352
0.0275	0.0319	0.0083	-0.0269	-0.0148	-0.0375	0.0215	0.0496
-0.0015	0.0315	-0.0322	-0.0342	-0.0417	-0.0514	-0.0117	0.0369
-0.0134	0.0167	-0.0325	-0.0142	-0.0321	-0.0316	-0.0220	0.0204
0.0017	-0.0320	-0.0495	0.0023	-0.0056	-0.0104	-0.0112	-0.0207

使用 NLSM Algorithm
陣列式麥克風個數：8
FIR 階數：5

處理後的資料
SNR=16.2286dB

SNR 比處理前增加
4.7685dB

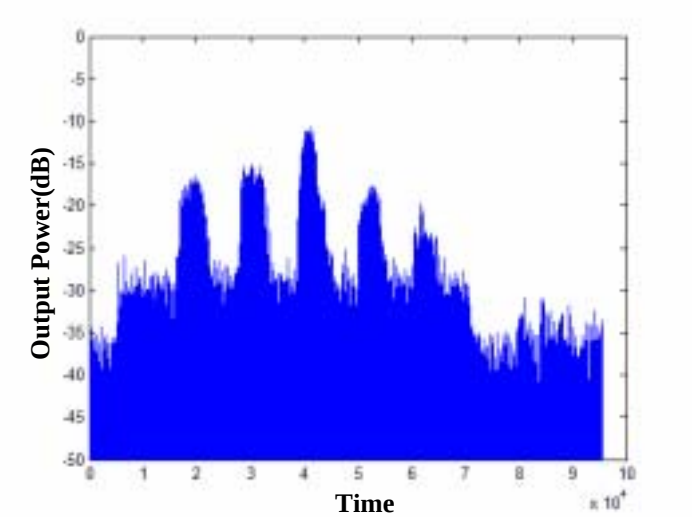

圖 29. 適應空間濾器輸出-麥克風個數 8、階數 5

係數：

0.1151	0.0343	0.1906	-0.0035	0.1337	-0.0203	0.0298	-0.1635
0.0230	0.0440	-0.0132	-0.0247	0.0777	0.0141	0.1342	-0.0156
0.0157	0.1144	-0.0725	-0.0854	0.0618	-0.0445	0.0006	0.1212
-0.0367	0.1399	-0.0728	-0.0109	0.0288	-0.0969	0.0314	0.1015
0.0398	0.1249	0.0472	0.0453	-0.0160	-0.1294	0.0236	-0.0401

使用 NLSM Algorithm
陣列式麥克風個數：8
FIR 階數：1

處理後的資料
SNR=16.5676dB

SNR 比處理前增加
5.1076dB

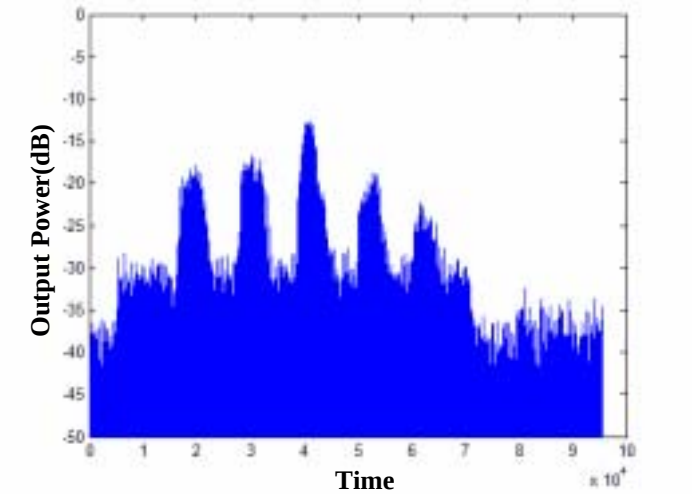

圖 30. 適應空間濾器輸出-麥克風個數 8、階數 1

係數：

0.1384	0.0089	0.2625	0.0837	0.1135	0.0041	0.1430	-0.0465

當 FIR 階數愈大時，增加的 SNR 愈多，但階數大到某一個範圍時 SNR 就不會再增加，同樣的當階數愈小時， SNR 隨著階數減少而減少，但當階數小於某一個範圍時，SNR 亦不會再下降。

4.2 實驗二、比較麥克風個數與 SNR 的關係

使用 NLSM Algorithm
陣列式麥克風個數：8
FIR 階數：10

處理後的資料
SNR=19.5390dB

SNR 比處理前增加
8.0790dB

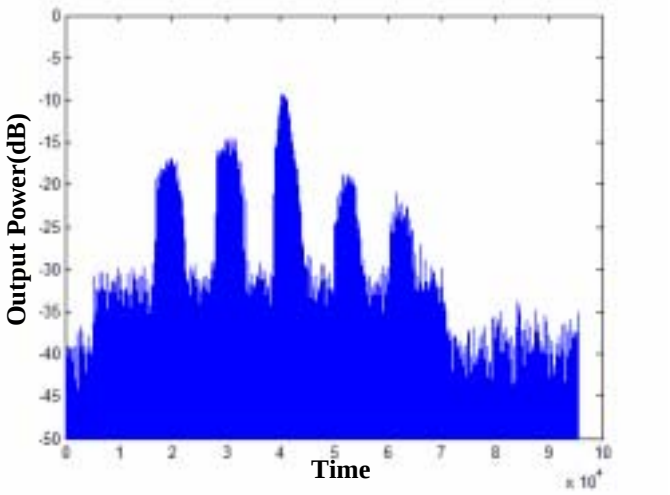

圖 31 適應空間濾器輸出-麥克風個數 8、階數 10

係數：

0.0624	0.0510	0.0811	0.0414	0.0772	0.0520	0.0083	-0.0730
0.0197	0.0389	0.0393	-0.0018	0.0330	0.0475	0.0503	-0.0007
0.0072	0.0235	0.0122	-0.0473	0.0149	-0.0024	0.0170	0.0450
-0.0065	-0.0080	-0.0137	-0.0202	0.0102	-0.0247	0.0059	0.0465
0.0081	-0.0027	0.0153	0.0051	0.0161	-0.0039	0.0254	0.0315
0.0316	0.0017	0.0320	-0.0037	0.0171	0.0001	0.0400	0.0352
0.0275	0.0319	0.0083	-0.0269	-0.0148	-0.0375	0.0215	0.0496
-0.0015	0.0315	-0.0322	-0.0342	-0.0417	-0.0514	-0.0117	0.0369
-0.0134	0.0167	-0.0325	-0.0142	-0.0321	-0.0316	-0.0220	0.0204
0.0017	-0.0320	-0.0495	0.0023	-0.0056	-0.0104	-0.0112	-0.0207

使用 NLSM Algorithm
陣列式麥克風個數：4
FIR 階數：10

處理後的資料
SNR=18.1903dB

SNR 比處理前增加
6.7303dB

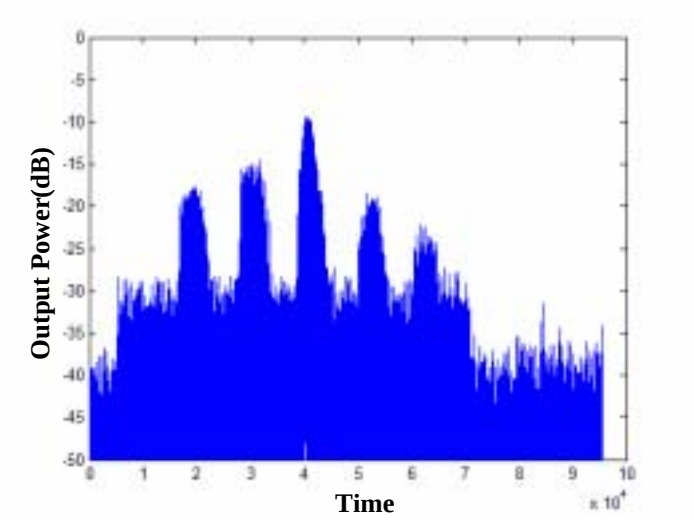

圖 32. 適應空間濾器輸出-麥克風個數 4、階數 10

係數：

0.1554	0.0657	0.1051	0.0612
0.0529	-0.0078	0.0608	0.0767
0.0250	-0.0470	0.0367	0.0406
-0.0004	-0.0255	0.0543	0.0228
0.0316	-0.0034	0.0620	0.0356
0.0154	-0.0187	0.0438	0.0372
-0.0324	-0.0240	0.0232	-0.0115
-0.0948	-0.0129	-0.0123	-0.0572
-0.0918	0.0127	-0.0044	-0.0393
-0.1032	0.0335	0.0507	-0.0637

使用 NLSM Algorithm
陣列式麥克風個數：2
FIR 階數：10

處理後的資料
SNR=16.7260dB

SNR 比處理前增加
5.2659dB

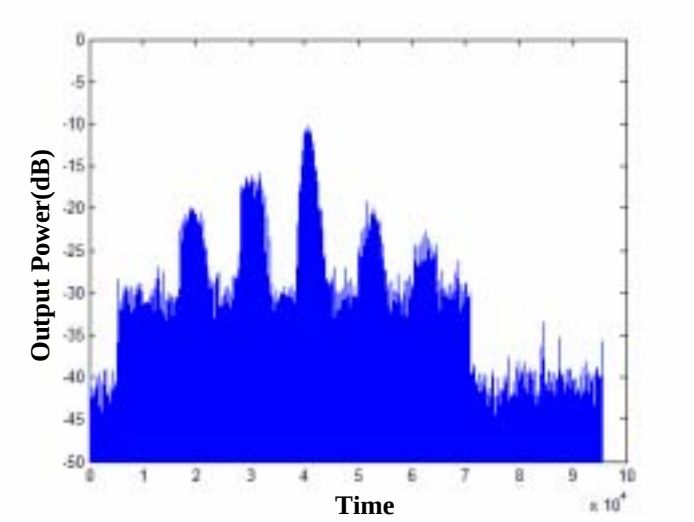

圖 33. 適應空間濾器輸出-麥克風個數 2、階數 10

係數：

0.2527	0.0777
-0.0189	0.1348
-0.0382	0.1165
-0.0161	0.0594
-0.0161	0.0733
-0.0400	0.0753
-0.0727	0.0012
-0.1019	-0.0681
-0.0320	-0.0325
0.0584	-0.1082

當麥克風的個數愈多，對於 SNR 的增加效果愈好，最主要的原因在於麥克風個數愈多，對於產生的 beamforming 的自由度就愈大，因此能將雜訊源方向的增益壓到很低，而語音訊號方向的增益仍能維持在一定的大小。當麥克風只剩兩個時，對於 SNR 的改善仍然能夠達到 5.2659dB，主要的原因在於目前的雜訊源只有一個，但當雜訊源超過兩個的時候，兩個麥克風的對於增加 SNR 的效果就會明顯變差。

Microphone Number	Order	Signal (dB)	Noise (dB)	SNR (dB)	Improve SNR (dB)
1	1	-18.8012	-30.2613	11.4601	0
8	10	-20.9207	-40.4597	19.5390	8.0790
4	10	-20.6620	-38.8524	18.1903	6.7303
2	10	-21.6847	-38.4107	16.7260	5.2659
8	5	-21.7027	-37.9313	16.2286	4.7685
4	5	-21.4106	-37.4978	16.0872	4.6272
2	5	-21.8621	-37.8480	15.9859	4.5258
8	1	-23.0120	-39.5797	16.5676	5.1076
4	1	-23.6932	-38.0371	14.3438	2.8838
2	1	-26.5263	-38.5457	12.0194	0.5593

表 2. 實驗數據整理

第五章 結論

1、使用陣列式麥克風設計空間濾波器，每一個麥克風之後接的適應性 FIR 濾波器係數最好可以達到 5 階以上，如此訊號的能量不容易因此衰減，處理前後能量的 SNR 可以提升約 10dB。

2、在干擾源愈多的情況之下，如果陣列式麥克風的個數不足，濾除雜訊的效果會愈差，最主要原因在於當麥克風個數愈多，設計空間濾波器的自由度愈大，對於不同方向進入的干擾源能夠壓低的程度也就愈好。

3、干擾源方向濾除雜訊的效果如果不夠好，可以在 NLMS 的輸入時加強干擾源的比重，如此可以將干擾源方向的增益壓得更低，使 SNR 增加，不過比重調整到一定大小之後 SNR 就不會提高。

4、對於近場語音訊號源，利用適應性訊號處理配合陣列式麥克風收到的具有空間特徵的訊號，設計空間濾波器的效果的確很好，最主要的原因在於適應性訊號處理在訊號處理的同時就已經包含了誤差的考量，即使元件匹配度不良、能量傳遞的損耗、電路的直流電壓位準偏移，設計出來的空間濾波器效果並不會因此而受到影響。

5、使用者判斷是否啟動 NLMS 改變空間濾波器的係數，如此當干擾源的位置移動時，空間濾波的效果會變得不好，需要再針對干擾源移動過後的環境再做一次 NLMS 改變空間濾波器的係數。其實 NLMS 也可以一直啟動著，根據變動的環境，不斷的更新空間濾波器的係數，只有使用者輸入語音訊號時才停止空間濾波器係數的更新，如此空間濾波器的係數既可配合環境不斷的變化，係數一直維持最適合目前環境的狀態，但是如何判斷語音訊號的輸入是一件相當不容易的事，因此有效率的判斷語音輸入的方法，是目前這個演算法中最需要再做研究的部分。

Reference

[1] Mattias Dahl, Ingvar Claesson, Sven Nordholm, and Sven Nordebo, "Acoustic Echo and Noise Cancelling Using Micphone Arrays," in *Proc. ISSPA, 1996*, pp. 379-382.

[2] Sven Nordholm, Ingvar Claesson, and Mattias Dahl, "Adaptive Microphone Array Employing Calibration Signals: An Analytical Evaluation," *IEEE Trans. on Speech and Audio Process*, vol. 7, pp. 241-252 , May 1999.

[3] S. Nordebo *et al*., "Noise Reducation Using an Adaptive Microphone Array for Speech Recognition in a Car," *in Proc. RVK93, Radio Vetenskaplig Konf*, Lund, Swedn, Apr. 1993.

[4] S.Nordholm, I. Claesson, and B. Bengtsson, "Adaptive Array Noise Suppression of Handsfree Speaker Input in Cars," *IEEE Trans. Veh. Technol.*, vol. 42, pp. 514-518, Oct. 1994.

[5] S.Nordholm, I. Claesson,and S. Nordebo, "Adaptive Beamforming:Spatial Filter Designed Blocking Matrix," *IEEE J. Oceanic Eng.,* vol. 19, no. 4, pp. 583-590, Oct 1994.

[6] Simon Doclo and Marc Moonen, "Design of Broadband Speech Beamformers Robust Against Errors in the Microphone Array Characteristics," in *Proc. ICASSP,* 2003, pp. v473-v476.

[7] Simon Doclo and Marc Moonen, "Design of Broadband Beamformers Robust Against Gain and Phase Errors in the Microphone Array Characteristics," *IEEE Trans. On Signal Processing,* vol. 51, pp. 2511-2526, Oct. 2003.

[8] Jan Madr de Hann, Nedelko Grbić, Ingvar Claesson, and Sven Eirk Nordholm, "Filter Bank Design for Subband Adaptive Microphone Arrays," *IEEE Trans. on Speech and Audio Processing*, vol. 11, pp. 14-23, Jan. 2003.

[9] Jean Barrére and Gilles Chabriel, "A Compact Sensor Array for Blind Sepration of Sources," *IEEE Trans. on Circuits and Systems-I: Fundamental Theory and Applications*, vol. 49, pp.565-574, May 2002.

[10] John R. Treichler, C. Richard Johnson, Jr., and Michael G. Larimore, *Theory and Design of Adaptive Filters,* Prentice-Hall, 2001.

[11] B. Widrow and S. D. Stearns,*Adaptive Signal Processing,* Prentice-Hall, 1985

[12] Simon Haykin, *Adaptive Filter Theory,* Fourth Edition, Prentice-Hall, 2001.

[13] Harry L. Van Trees, *Optimum Array Processing*, 2002.

[14] R. Monzingo and T. Miller, *Introduction to Adaptive Array*, 1980.

[15] Bradley Bargen and Peter Donnelly, *Inside DirectX*, 1999, pp. 203-246.