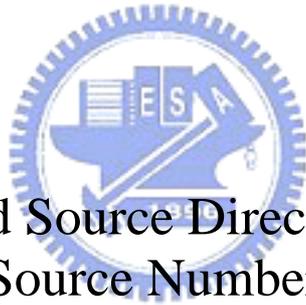


國立交通大學  
電機與控制工程研究所

碩士論文

多聲源方位偵測與聲源數量估算



Multiple Sound Source Direction Estimation and  
Sound Source Number Estimation

研究生：汪正剛  
指導教授：胡竹生 博士  
楊谷洋 博士

中華民國九十七年七月

# 多聲源方位偵測與聲源數量估算

## Multiple Sound Source Direction Estimation and Sound Source Number Estimation

研究生：汪 正 剛

Student : Cheng-Kang Wang

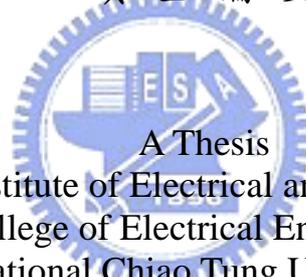
指導教授：胡 竹 生 博士

Advisor : Dr. Jwu-Sheng Hu

楊 谷 洋 博士

Dr. Kuu-Young Young

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  
National Chiao Tung University  
in partial Fulfillment of the Requirements  
for the Degree of Master  
in

Electrical and Control Engineering

July 2007

Hsinchu, Taiwan, Republic of China

中華民國九十六年七月

# 多聲源方位偵測與聲源數量估算方法

研究生：汪 正 剛

指導教授：胡 竹 生 博士  
楊 谷 洋 博士

國立交通大學電機與控制工程研究所碩士班

## 摘 要



本論文提出一套在聲速以及聲源個數皆未知的情況下藉由分散式的麥克風陣列擷取到的聲音訊號估算聲源個數以及聲源方向的演算法。並在演算法中提出了一套結合特徵結構分析(Eigenstructure)和廣義互相關性演算法(GCC, Generalized Cross-Correlation)的時間延遲估算演算法(ES-GCC)。ES-GCC 利用聲場中的特徵向量估算麥克風之間的時間延遲關係，論文中將說明 ES-GCC 較一般 GCC 來的穩定。在計算完時間延遲關係後，本論文利用最小平方法(Least square)估算出聲源方位以及目前的聲速。而在多聲源的情形下，藉由判斷聲速估算的合理性可以排除不正確的聲源角度，再經由累積估算結果並且套用 Adaptive K-means++演算法，即可估算出聲源個數以及聲源角度。本論文實驗皆於真實環境中執行，用以測試本論文所提演算法的效能。

# Estimation of the Source Number and Directions of Arrival in a Multiple Sound Source Environment

Student : Cheng-Kang Wang

Advisor : Dr. Jwu-Sheng Hu  
Dr. Kuu-Young Young

Institute of Electrical and Control Engineering

## Abstract

This work proposes a multiple sound source direction estimation and sound source number estimation method by using distributed microphone array without the information of sound velocity. This work also proposes an eigenstructure-based generalized cross correlation method (ES-GCC) for estimating time delay between microphones. Upon obtaining the time delay information, the sound source direction and velocity can be estimated by least square method. In multiple sound source case, the time delay combination among microphones is arranged such that the estimated sound speed value falls within an acceptable range. By accumulating the estimation result of sound source direction and using adaptive K-means++ algorithm, we can estimate the sound source number and direction for each source. Experimental results are carried out in the real environment to evaluate the performance of the proposed algorithm.

# 誌 謝

經過兩年碩士生涯自己的努力以及眾多朋友和師長的幫助下，終於到了學生生涯的終點，真的很感謝胡竹生博士和楊谷洋博士兩年來的指導，老師的指導讓我在研究的路上能夠保持正確的方向，也讓我學習用嚴謹的態度面對眼前的任何問題，而也在老師們的指導下使得本論文得以順利完成。

在實驗室的兩年裡，因為有學長姐、同學以及學弟妹的陪伴，讓我的研究生生涯過的相當快樂，感謝立偉學長在我大學時期帶領我接觸了機器人的研究，感謝劉大以及興哥帶領我進入聲音研究的領域，是你們啟發我研究的興趣，感謝興哥有耐心的在研究以及做計劃各方面幫助我讓我可以完成各項挑戰，感謝幽默逗趣的宗敏學長，感謝鏗元學姊不定時的咖啡時間讓我保持清醒的頭腦，感謝永融給我許多研究以及實作方面的指導，感謝大師兄讓我看到了認真負責的做事態度；而和我一起奮鬥的同學們也讓我的研究之路不是孤獨一人的：考試報告界的王者阿吉，邏輯很強很有型的 HCY，永遠的樂天派以及永遠的活動組長槓葛葛，好像很迷糊但是卻是同學中最認真的瓊文，俗又有力愛車成癡的俊宇，做事認真的車神明唐；活潑的學弟們也讓實驗室的氣氛活躍了起來：英文超強說話有調理的嘟嘟，造型相當有特色的追星族 LUNDY，籃球狂熱的肉鬆，還有拳霸 JUDO；當然還要感謝什麼事都要做的實驗室助理淑玲；感謝大家和我一起走過研究生兩年的生涯，這段回憶我會好好珍藏一輩子的。

還要特別感謝聲音研究的小組，SIPP 小隊的成員們：興哥、HCY、明唐，研究所生涯中有和這樣的團隊一起工作過已經沒有什麼遺憾，希望實驗室聲音研究的小組以後也能夠繼續成長茁壯。

另外，也謝謝每天和我一起打排球的朋友們，你們讓我在新竹六年的回憶不只有研究，讓我很快樂的度過每一天；最後要感謝我的父母和家人，是你們的鼓勵與幫助讓我無後顧之憂的專心學習，完成我碩士的學業。

最後，僅以本論文向家人獻上最誠摯的謝意。

# 目錄

摘要 .....	i
Abstract .....	ii
誌謝 .....	iii
表列 .....	vi
圖列 .....	vii
<b>第一章 緒論 .....</b>	<b>1</b>
1.1 研究動機 .....	1
1.2 本研究創新說明 .....	2
1.3 論文架構 .....	2
<b>第二章 Direction of arrival(DOA) estimation 簡介 .....</b>	<b>3</b>
2.1 DOA 簡介 .....	3
2.2 TDE 與 Eigenspace Method 簡介 .....	3
2.2.1 TDE(Time Delay Estimation) 簡介 .....	3
2.2.2 Eigenspace Method 簡介 .....	6
2.3 結合 Eigenspace Method 和 GCC 之時間延遲估算演算法 .....	7
<b>第三章 ES-GCC 演算法 .....</b>	<b>8</b>
3.1 簡介 .....	8
3.2 GCC 演算法 .....	8
3.3 ES-GCC 演算法 .....	10
3.4 ES-GCC 與 PHAT-GCC 之比較 .....	13
<b>第四章 聲源方位以及聲速估測演算法 .....</b>	<b>15</b>
4.1 簡介 .....	15
4.2 聲源方位以及聲速估測原理 .....	15
4.2.1 聲源距離對於方位以及聲速估測的影響 .....	15
4.2.2 利用最小平方法求解以及麥克風陣列幾何形狀的影響 .....	17
4.2.3 估測誤差分析 .....	20
4.3 聲源方位估測結果 .....	25

<b>第五章</b>	<b>多聲源方位偵測與聲源數量估算演算法 .....</b>	<b>26</b>
5.1	簡介 .....	26
5.2	多聲源方位估測演算法 .....	26
5.3	<i>Adaptive K-means++</i> 演算法 .....	30
5.3.1	<i>K-means++</i> 演算法 .....	30
5.3.2	<i>Adaptive K-means++</i> 演算法 .....	31
5.4	多聲源方位偵測與聲源數量估測結果 .....	33
<b>第六章</b>	<b>未來展望 .....</b>	<b>35</b>
<b>Reference:</b>	<b>.....</b>	<b>36</b>



# 表 列

表 一、聲源方位估算結果 .....	25
表 二、時間延遲量組合.....	28
表 三、單聲源方位與聲源數量估算結果 .....	33
表 四、雙聲源方位與聲源數量估算結果 .....	33
表 五、同時四聲源方位與聲源數量估測結果 .....	34



# 圖 列

圖 一、線型陣列模型 .....	11
圖 二、(A)時間延遲估計量(PHAT-GCC)(B)時間延遲估計量(ES-GCC).....	13
圖 三、(A)雙聲源時間延遲估計量(PHAT-GCC)(B) 雙聲源時間延遲估計量 (ES-GCC).....	13
圖 四、平行波與兩個麥克風之間的幾何關係示意圖 .....	16
圖 五、SAMPLE RATE 對誤差的影響.....	20
圖 六、聲源距離對誤差向量大小的影響 .....	21
圖 七、聲源距離對聲速估測的影響 .....	21
圖 八、聲源角度與誤差大小之關係 .....	22
圖 九、SAMPLE RATE 對位置估測的影響.....	23
圖 十、聲源距離與位置估測誤差關係 .....	23
圖 十一、麥克風數量對誤差的影響 .....	24
圖 十二、麥克風陣列與聲源相對位置示意圖 .....	25
圖 十三、多聲源 ES-GCC 結果圖 .....	26
圖 十四、與相關性最大值的 0.9 倍比較後的 ES-GCC 結果圖 .....	27
圖 十五、空間中雙聲源與麥克風對之位置關係圖 .....	27
圖 十六、估算聲源角度分布(A)單聲源(B)雙聲源(C)四聲源.....	29
圖 十七、ADAPTIVE K-MEANS++演算法流程圖.....	31
圖 十八、聲源距離與角度估算誤差關係圖 .....	32
圖 十九、麥克風陣列.....	33

# 第一章 緒論

## 1.1 研究動機

估測聲音訊號在空間中的方位，或者是入射方向與感測裝置之間的關係，此類研究一般來說稱為 Direction of arrival (DOA) estimation。DOA 的技術層面巨觀上可以分成兩大類，第一類是 TDE (Time Delay Estimation)，第二類是 Eigenspace Method。TDE 的方法都只有針對一個聲音來源來作方位的辨別，沒有辦法在同一時間內辨別出數個聲源的位置。如果要同時辨別數個聲源的方位就必須要使用 Eigenspace Method，但是 Eigenspace Method 需要知道總聲源數量才可以辨別多個聲源的方向。

TDE 的方法通常都是使用兩顆麥克風，因為聲源從不同的方位傳遞到這兩顆麥克風都會造成不同的響應，此法便是利用聲音從空間中傳遞的特性所造成的時間差來作方位的估計。也因為只有使用兩顆麥克風，所以假設從聲源傳遞過來的聲音訊號是平面波是相當合理的。此類方法最常使用的量測標準就是 GCC (Generalized Cross-Correlation)，藉由找出這個關係式的最大值就會等效於找到了聲音傳遞到這兩顆麥克風的時間差，在經由時間差的資訊便可以推測出聲源相對於這兩顆麥克風的方位了。而 Eigenspace Method 則是另一種偵測訊號源方向的方式，此類方法利用不同訊號間特徵向量的分布關係，以互相投影的方式來估計出訊號的方向，因此這類方法可以在聲源數量已知的情況下同時對多個聲源同時加以估計。

以上兩種聲源方位估測的方法都必須在聲速和聲源數量已知的情況下使用，對於不同環境下聲速的變化以及聲源數量未知的情況無法有效的估算聲源數量以及聲源方位。在論文中，我們提出了一套在聲速以及聲源數量未知的情況下，可以同時估算聲速、聲源數量以及聲源方位的演算法，此演算法結合了傳統 TDE 和 Eigenspace Method 兩者的優點創立一套較為穩定的 TDE 演算法 ES-GCC (Eigenstructure-based Generalized Cross-correlation Method)，並且使用簡單的最小平方法 (Least square) 估算聲速和聲源方向，同時利用 Adaptive K-means++

演算法估算聲源數量。

## 1.2 本研究創新說明

本論文對聲源方位偵測與聲源數量估算提出如下的創新：

1. 結合特徵結構分析(Eigenstructure)以及傳統時間延遲估算演算法(GCC)開發出一套較為穩定的時間延遲估算演算法。
2. 利用聲源位置和麥克風陣列結構之關係和最小平方法(Least square)推導出一套可以同時估算聲源方向以及聲速的演算法。
3. 利用聲速估算的正確性以及使用 Adaptive K-means++演算法建立一套聲源個數估算的演算法。

## 1.3 論文架構

本篇論文包含了三個主要的部份，分別是時間延遲估算演算法(ES-GCC)的理論、如何利用最小平方法(Least square)估算聲速和聲源方位、以及如何利用聲速估算的結果結合 Adaptive K-means++演算法作聲源個數的估計。底下將大致描述各章節的主要內容：

第二章：將介紹 Direction of arrival (DOA) estimation 的概念、時間延遲估算(Time Delay Estimation)、傳統廣義互相關演算法(GCC, Generalized Cross-Correlation)、以及 Eigenspace Method。

第三章：介紹結合特徵結構分析(Eigenstructure)和廣義互相關性演算法的時間延遲估算演算法(ES-GCC), 並且比較 ES-GCC 和傳統 GCC 兩者之不同處。

第四章：說明如何利用麥克風陣列結構和聲源的位置關係和最小平方法(Least square)估算聲源方位以及聲速，並且對不同的聲波假設下的情況(平面波，球面波)做比較

第五章：將說明如何藉由聲速估算的正確性在多聲源的情況計算出聲源數量以及聲源方位，簡介 Adaptive K-means++演算法，以及如何將其應用在聲源個數估算上

第六章：未來的展望

## 第二章 Direction of arrival(DOA) estimation 簡介

### 2.1 DOA 簡介

估測訊號在空間中的方位，或者是入射方向與感測裝置之間的關係，此類研究一般來說稱為 Direction of arrival (DOA) estimation，DOA 的技術層面巨觀上可以分成兩大類，第一類是 TDE (Time Delay Estimation)，第二類是 Eigenspace Method。TDE 的方法都只有針對一個聲音來源來做方位的辨別，沒有辦法在同一時間內辨別出數個聲源的位置，如果要同時辨別數個聲源的方位就必須要使用 Eigenspace Method，但是 TDE 的好處是只需要使用兩顆麥克風，所需的硬體架構較為簡單，並且其運算量相較之下也比較符合即時應用的要求。而 Eigenspace Method 的好處就是可以在總聲源數量已知的前提下辨別數個聲源，不會因為有數個聲源同時存在而發生無法判別的情況。所以使用者可以根據不同應用的需求來選取其中的一種，甚至可以兩者並行處理。

### 2.2 TDE 與 Eigenspace Method 簡介

#### 2.2.1 TDE(Time Delay Estimation)簡介

這一類的方法通常都是使用兩顆麥克風，因為聲源從不同的方位傳遞到這兩顆麥克風都會造成不同的響應，此法便是利用聲音從空間中傳遞的特性所造成的時間差來作方位的估計。也因為只有使用兩顆麥克風，所以假設從聲源傳遞過來的聲音訊號是平面波是相當合理的。此類方法最常使用的量測標準就是 GCC(Generalized Cross-Correlation)，藉由找出這個關係式的最大值就會等效於找到了聲音傳遞到這兩顆麥克風的時間差，在藉由時間差的資訊便可以推測出聲源相對於這兩顆麥克風的方位了。而 GCC 的定義卻延伸出相當多的變化，也是推動此類方法持續演進的關鍵技術，這類方法都是轉換到頻率域，並且定義出 Weighting function，這相關的研究早在 1970 年代便有很多知名的方法產生，如 1971 年 Roth[1]所提出的 Roth Processor，此方法的好處在於可以壓低第一顆麥克風電子雜訊和空間雜訊的所在頻率所產生的影響，進而增加時間差估計的正確性。而接下來在 1972 年 Carter 等人[2]提出了名為 PHAT(The Phase Transform)的

方法，如果兩顆麥克風彼此的雜訊的分佈情況是無關的，且所估計出來的 Cross Correlation 和真正的 Cross Correlation 同樣時，此法可以有相當好的效果，但是其錄音時的 SNR 必須要相當高也是其要求之一。時間又往前推演到 1973 年 Carter 等人[3]又提出了名為 SCOT(The Smoothed Coherence Transform)的方法，因為 Roth 只考慮到第一顆麥克風的雜訊所造成的影響，SCOT 則是同時去考慮到兩顆麥克風的雜訊所造成的影響，也就是說 Roth 只是 SCOT 的某個特殊例子，如此可以更增進時間估計的正確性。

然而高的 SNR 的需求仍然是相當重要，一直到 1976 年 Knapp 以及 Carter 又提出了 ML(Maximum Likelihood)的方法[4]才比較放鬆了高 SNR 這樣的限制，在此方法中是先用 Eckart Filter 作為定義 Weighting function 的基礎，並且假設雜訊以及聲源的統計分佈都是高斯分佈(Gaussian)，最後推演出新的 Weighting function，而此方法也一直被廣泛的使用。但是上述的方法都沒有辦法對付有反射的情況，這是由於上述方法都是以單一入射路徑(Single Path)為考量前提所推演出的理論，如此若運用到真正的情況將會變得沒有這麼適用，在 1996 年 Champagne 等人[5]利用模擬不同的反射時間，進而觀察上述眾多方法的效能，可以從其文獻中看出如果反射時間大於 0.15 秒時則估計出來的結果將開始呈指數性的變差，所以上述方法僅能使用在反射較小或者空間較大的環境。如此的反射限制在 1995 年同樣是 Champagne 等人[6]提出了 GCC-CEP 的方法來作出解決，在此篇文獻當中提出了一個 Prefilter 把空間中的 channel 效應作消除，然後再運用 ML 的方法，而此 Prefiler 則是轉換到 Cepstral Domain 中求出，並且利用 Homomorphic Deconvolution 的概念直接減除，如此可以增加對於反射的抵抗能力。但是 channel 的估計方法則是必須要根據不同的環境而有不同的參數必須要作出調整，並沒有一個相當理論上的方法來告知如何調整其參數，必且如果要有好的效能則求取此 Prefilter 所需的內容也必須要作調整。如果運用的領域是使用者會移動的情況下則不容易有好的效能。

因此在 1997 年 Brandstein 等人[7]又提出了較新的方法，此方法並不要求取 channel 的響應，而是利用新的統計函數(Tukey's Biweight)來重新定義

Weighting function，並且推演出對於反射情況下更穩健的估測方法，雖然在效能上有不錯的改進，但是也有個參數會影響到不同環境下的效能表現，因此必須先前去作一些簡單的測試才可以運用到真實的環境。

上述所有的方法都是針對 1-D 來估算出聲源相對於麥克風的方位。除此之外也有相關的 2-D 研究提出，如在 1997 年 Omologo 等人[8]提出了用兩組麥克風對，亦即使用四顆麥克風來求出聲源的 2-D 座標，而 Weighting function 的是利用 CSP(Cross-power spectrum phase)重新作定義，並且配合 CM (Coherence Measure) 的方法來求出方位角，最後將這兩組麥克風對各自求出的方位角取出，用以找出聲源的 2-D 方位，從實驗結果可以觀察出當聲源和麥克風的距離大於 100 公分以上後，誤差會快速的變小，但是若大於 140 公分以上誤差又會開始有變大的傾向。除此之外，我們也在先前的研究中[9][10]利用 Cross-Correlation 的方法求取出重點頻率單位，進而利用 GCC 求出相對的方位角，最後利用麥克風對本身的排列特性求取出聲源的 2-D 方位。

雖然反射的問題已經影響變得不大，但是如果在偵測聲源方位的同時，突然有雜訊發生，則會嚴重影響估測的準確性，必且其影響到的那段期間都無法正確的估算出聲源的方位，為了避免這樣的問題，也有些研究[11][12]便針對之前雜訊模型的高斯假設作出更改，引進了 Nikias 等人[13]所提出的 Alpha-stable 的分佈來 model 雜訊，並且發現針對語音應用的例子可以使用  $S\alpha S$  (Symmetric Alpha Stable) 來 model 雜訊，其中  $0 < \alpha \leq 2$ 。當  $\alpha = 2$  時， $S\alpha S$  也就等效成高斯分佈，亦即高斯分佈只是  $S\alpha S$  的某個特殊例子。從文獻中也指出如此的假設會更貼近真正環境下雜訊的分佈狀況，除此之外還一併提出了新的計算 Cross-power 的方法，稱為 FLOS(Fractional lower order statistics)，這屬於非線性的求取方式，兩者互相配合後可以發現便會對於突然發生的雜訊有很高的抗干擾能力，只會在相當短暫的時間內受到干擾，且其干擾所造成的估算錯誤也不會誤差過多，再加上此一方法並不需要事先知道雜訊以及聲源的特性來加以輔助計算，如此種種優點便使得此理論相當適合運用於實際的環境。

## 2.2.2 Eigenspace Method 簡介

Eigenstructure Direction-of-Arrival (DOA) Estimation 則是另一種偵測訊號源方向的方式，此類方法利用不同訊號間特徵向量的分布關係，以互相投影的方式來估計出訊號的方向，因此這類方法可以同時對多個聲源同時加以估計。雖然此類 Subspace-base 的 DOA Estimation 方法在窄頻訊號方面已有大量文獻討論，然而，相對於通訊方面的窄頻訊號，語音訊號乃是寬頻假設，也因此其估測難度也較高，同時，關於寬頻訊號的論文卻較為不足。

於 1984 年左右，Wax 等人[14]將窄頻所採用的 DOA Estimation 演算法延伸至寬頻訊號上，其主要觀念乃是將原本窄頻訊號的 Time-domain eigenstructure method 延伸成為 Frequency-domain eigenstructure method。而 H.Wang 和 M. Kaveh[15]則是提出了一種考慮了 Coherent Source 情況的 Wide-band DOA 演算法，其主要目的是為了解決前面所提出的 Eigenstructure 方法在有嚴重反射而形成的 Coherent Source 時，會發生 Desired signal cancellation 效應，使得其穩健度下降的問題，並同時也提出一種稱為 Coherent minimum akaike information criterion estimation(MAICE)來估計訊號源的個數。而另一方面，Bienvenu[16]、Buckley 和 Griffiths[17]等人亦提出了將空-時域訊號所觀察的空間轉換為訊號-雜訊空間的方法，並提出相對的搜尋方式來估測 DOA。

到了 90 年代，Doron[18]等人提出了一些以 Maximum likelihood(ML) estimation 為基礎的 DOA search function，然而，此類方式卻因為 Search function 需要大量的運算資源而顯得不夠實際，因此這方面的研究也引進了 Bresler 和 Macovski[19]所提出的 Narrow-band iterative quadratic maximum likelihood(IQML) 來實現前述的 ML。Agrawal 和 Prasad 整合了前述的觀念與 Harmonic Source Model 而設計出一種可同時應用於 Non-coherent 和 Coherent source 的 DOA estimator[20]，而 Messer[21]則研究了各種不同以 Eigenstructure 為基礎的 DOA estimation 方法的極限，說明了頻譜資訊對此類 Eigenstructure based DOA 估測方式的幫助有一定的限制，因此 Agrawal 和 Prasad[22]又提出了一種不需要頻譜資訊，而僅以空間模型為主的 Broadband DOA estimation 演算法，並證明他們對

訊號所作的 Flat power spectrum 假設對其估測結果的影響有限。

上述的演算法皆在 Far field 情況下討論，在此前提下，聲音的傳遞被視為平面波，使得空間對麥克風陣列效應的模型可以以整數倍的延遲關係簡化，然而在某些使用中，聲源的對麥克風陣列的距離相對於麥克風陣列孔徑相對甚大時，討論的 DOA 問題就不再符合平面波假設，因此對這類的問題，亦有 Near field 假設的 DOA estimation 演算法加以討論。Lee 等人[23]即利用重建一組 Far-field approximation (FFA) covariance matrix 的方式，使得原始的 Eigenstructure DOA 演算法可以直接套用至近似後的 covariance matrix 上，並計算出較為正確的 DOA 結果。

### 2.3 結合 Eigenspace Method 和 GCC 之時間延遲估算演算法

經由前面章節的介紹，我們可以對 Eigenspace Method 和 GCC 兩者之特色有初步的了解，而本篇論文及提出之時間延遲估算演算法則是結合兩者之優點，使得麥克風之間的時間延遲量可以估算的更穩定。

首先藉由 Eigenspace Method 可以將聲音訊號分為訊號空間和雜訊空間的特色，將麥克風陣列收到的聲音訊號中主要的訊號分布求出，這樣做的用意是可以壓低雜訊分布對於後段處理的影響，接者再使用 GCC 求出麥克風之間的時間延遲量，在第三章將有較詳細的推導以及實驗數據說明本論文提出之時間延遲估算演算法比傳統的演算法來的穩定。

## 第三章 ES-GCC 演算法

### 3.1 簡介

傳統 GCC 的作法是直接對麥克風收到的聲音訊號做計算，這樣的作法在沒有雜訊干擾的情況下相當的直觀，但是實際的狀況是空間中會有許多的雜訊干擾，反射、折射對於麥克風之間時間延遲的估計都會有相當大的影響。使用麥克風陣列提升麥克風數量後，訊號分布的緯度也增加了，此時就可以套用 Eigenspace Method 的觀念，找出聲音訊號在麥克風陣列中的主要分布，而先找出主要分布再利用 GCC 計算麥克風之間的時間延遲結果也會比較穩定，以下將先介紹 GCC 演算法的內容，然後再說明 ES-GCC 演算法。

### 3.2 GCC 演算法

首先假設空間中有一個聲源，在理想情況下兩個麥克風收到的聲音訊號可以分別表示為：

$$x_1(t) = s_1(t) + n_1(t) \quad (3.2.1)$$

$$x_2(t) = \alpha s_1(t + D) + n_2(t) \quad (3.2.2)$$

其中假設  $s_1(t)$ ， $n_1(t)$  和  $n_2(t)$  都是 WSS，並且  $s_1(t)$  和  $n_1(t)$  以及  $n_2(t)$  是 uncorrelated。其中 D 就是真正的 Delay，而  $\alpha$  代表的是改變大小的 Scale value，並且其 D 和  $\alpha$  的改變相較於  $s_1(t)$  是緩慢變化的，而此時麥克風之間的 cross correlation 可以表示為：

$$R_{x_1, x_2}(\tau) = E[x_1(t)x_2(t-\tau)] \quad (3.2.3)$$

其中  $E$  代表期望值，而使得(3.2.3)達到最大值的  $\tau$  即為兩個麥克風之間的時間延遲量，因為實際觀測的時間是有限的，所以 cross correlation 的估算可以表示為：

$$\hat{R}_{x_1, x_2}(\tau) = \frac{1}{T-\tau} \int_{\tau}^T x_1(t)x_2(t-\tau)dt \quad (3.2.4)$$

其中 T 代表觀測的時間間隔，而 cross correlation 和 cross power spectrum 之間的關係可以用 Fourier Transform 表示成下面這個形式：

$$R_{x_1, x_2}(\tau) = \int_{-\infty}^{\infty} G_{x_1, x_2}(f)e^{j2\pi f\tau} df \quad (3.2.5)$$

現在來考慮實際空間的狀態，麥克風收到的聲音訊號是經過空間轉換後的狀態，

因此實際麥克風之間的 cross power spectrum 可以表示為：

$$G_{y_1, y_2}(f) = H_1(f)H_2^*(f)G_{x_1, x_2}(f) \quad (3.2.6)$$

其中， $H_1(f)$  以及  $H_2(f)$  分別代表聲源到第一個麥克風以及第二個麥克風的空間轉換函數，因此我們定義麥克風對之間的 Generalized correlation 為：

$$R_{y_1, y_2}^{(g)}(\tau) = \int_{-\infty}^{\infty} \psi_g(f)G_{x_1, x_2}(f)e^{j2\pi f\tau} df \quad (3.2.7)$$

其中

$$\psi_g(f) = H_1(f)H_2^*(f) \quad (3.2.8)$$

實際上，在有限的時間觀察下我們只能以估算的  $\hat{G}_{x_1, x_2}(f)$  取代  $G_{x_1, x_2}(f)$ ，因此(3.2.7)式改寫為：

$$\hat{R}_{y_1, y_2}^{(g)}(\tau) = \int_{-\infty}^{\infty} \psi_g(f)\hat{G}_{x_1, x_2}(f)e^{j2\pi f\tau} df \quad (3.2.9)$$

利用(3.2.9)式，我們就可以估算麥克風對之間的時間延遲量，而  $\psi_g(f)$  的選取對於時間延遲的估計也有一定的影響，而在本篇論文中使用的是由 Carter 等人提出的名為 PHAT(The Phase Transform)的方法，也就是：

$$\psi_g(f) = \frac{1}{|G_{x_1, x_2}(f)|} \quad (3.2.10)$$

此方法在兩個麥克風彼此的雜訊分布是無關的時候，有相當好的效果。

### 3.3 ES-GCC 演算法

首先假設一個有  $M$  個麥克風的麥克風陣列，並且同時有  $d$  個聲源存在，此時第  $m$  個麥克風收到的聲音訊號可以表示為：

$$x_m(t) = \sum_{k=1}^d a_{mk} s_k(t - \tau_{mk}) + n_m(t) \quad (3.3.1)$$

其中， $a_{mk}$  代表第  $k$  個 source 到第  $m$  個麥克風的訊號大小改變量， $n_m(t)$  代表了第  $m$  個麥克風收到的雜訊，對(3.3.1)式做傅立葉轉換後可以改寫為：

$$X_m(\omega_f, k) = \sum_{k=1}^d a_{mk} S_k(\omega_f, k) e^{-j\omega_f \tau_{mk}} + N_m(\omega_f, k) \quad (3.3.2)$$

$f = 1, 2, \dots, F$

其中， $\omega_f$  代表觀測的頻率範圍，而  $k$  代表不同的時間間隔，將(12)式重新以矩陣的形式改寫：

$$\mathbf{X}(\omega_f, k) = \mathbf{A}(\omega_f) \mathbf{S}(\omega_f, k) + \mathbf{N}(\omega_f, k) \quad (3.3.3)$$

其中

$$\mathbf{X}^T(\omega_f, k) = [X_1(\omega_f, k), \dots, X_M(\omega_f, k)] \quad (3.3.4)$$

$$\mathbf{N}^T(\omega_f, k) = [N_1(\omega_f, k), \dots, N_M(\omega_f, k)] \quad (3.3.5)$$

$$\mathbf{S}^T(\omega_f, k) = [S_1(\omega_f, k), \dots, S_d(\omega_f, k)] \quad (3.3.6)$$

$$\mathbf{A}(\omega_f) = \begin{bmatrix} a_{11} e^{-j\omega_f \tau_{11}} & \dots & a_{1d} e^{-j\omega_f \tau_{1d}} \\ \vdots & & \vdots \\ a_{M1} e^{-j\omega_f \tau_{M1}} & \dots & a_{Md} e^{-j\omega_f \tau_{Md}} \end{bmatrix} \quad (3.3.7)$$

接著我們計算麥克風陣列收到的訊號之 correlation matrix 並且做 eigenvalue decomposition，此式可以表示為：

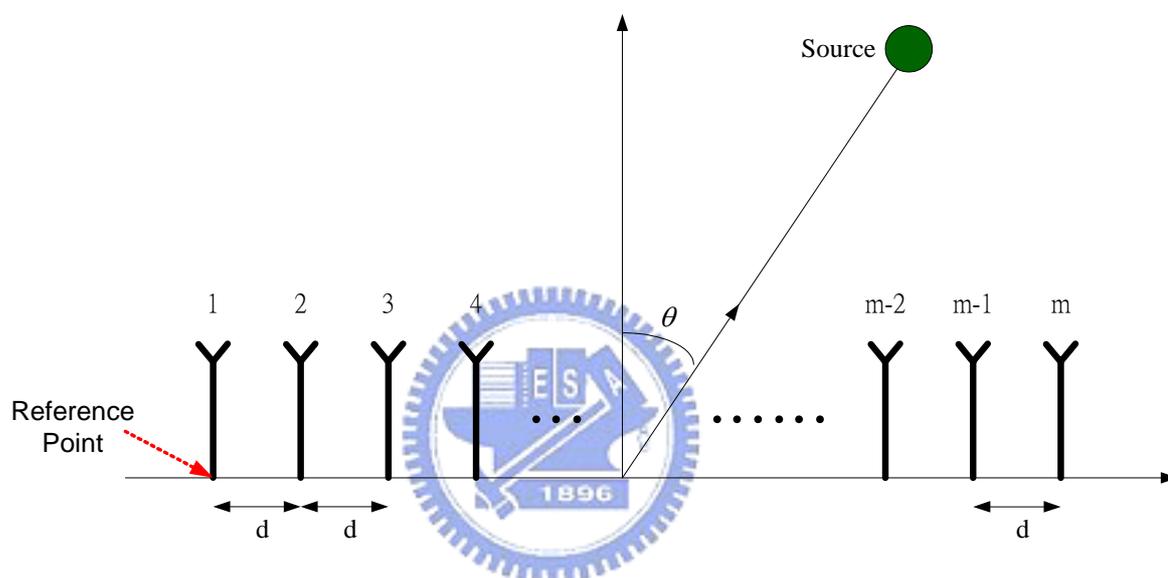
$$\begin{aligned} \mathbf{R}_{xx}(\omega_f) &= \frac{1}{N} \sum_{k=1}^N \mathbf{X}(\omega_f, k) \mathbf{X}^H(\omega_f, k) \\ &= \sum_{i=1}^M \lambda_i(\omega_f) \mathbf{V}_i(\omega_f) \mathbf{V}_i^H(\omega_f) \end{aligned} \quad (3.3.8)$$

其中， $N$  代表計算 correlation matrix 的音框數(frame number)， $\lambda_i(\omega_f)$  和  $\mathbf{V}_i(\omega_f)$  分別是 eigenvalue 以及相對應的 eigenvector，而  $\lambda_1(\omega_f) \geq \lambda_2(\omega_f) \geq \dots \geq \lambda_M(\omega_f)$ 。

在 Multiple Signals Classification Method (MUSIC)演算法[24]中將經由(3.3.8)式轉換後的 eigenvector 分為兩群：

1.  $V_1(\omega_f), V_2(\omega_f) \cdots V_d(\omega_f)$  稱為訊號的特徵向量並且以  $\text{span}\{V_1(\omega_f), V_2(\omega_f) \cdots V_d(\omega_f)\}$  為訊號子空間。
2.  $V_{d+1}(\omega_f), V_{d+2}(\omega_f) \cdots V_M(\omega_f)$  稱為雜訊的特徵向量並且以  $\text{span}\{V_{d+1}(\omega_f), V_{d+2}(\omega_f) \cdots V_M(\omega_f)\}$  為訊號子空間。

接著 MUSIC 演算法利用聲場特性與陣列幾何形狀的關係建立了 array manifold vector，以均勻線性陣列為例，如下圖一所示：



圖一、線型陣列模型

若設定第一個麥克風為參考點，則此線型陣列的 array manifold vector 定義為：

$$a^T(\theta) = [1 \ e^{j*k_c*d*\sin\theta} \ \dots \ e^{j*k_c*d*(m-1)*\sin\theta}] \quad (3.3.9)$$

其中  $k_c = 2\pi / \lambda_c$  表示波數(wavenumber)， $\lambda_c$  表示波長， $d$  是兩麥克風之間的距離， $m$  則代表麥克風的個數，而 array manifold vector 中第  $i$  項就代表了第  $i$  個麥克風與第一個麥克風在聲源方向為  $\theta$  的時候的相位差，相位差相對於時間領域(time domain)即為時間延遲量差，因此將 array manifold vector 轉為時間領域可以寫成：

$$a_{time\ domain}^T(\theta) = [1 \ \frac{d}{\lambda_c}*\sin\theta \ \dots \ \frac{d}{\lambda_c}*(m-1)*\sin\theta] \quad (3.3.10)$$

MUSIC 演算法利用 array manifold vector 與  $V_{d+1}(\omega_f), V_{d+2}(\omega_f) \cdots V_M(\omega_f)$  內積的結果判斷聲源的方向，其公式為：

$$\theta_{DOA} = \max \left( \frac{1}{a^H(\theta) \left( \sum_{i=d+1}^M V_i V_i^H \right) a(\theta)} \right) \quad (3.3.11)$$

由於訊號子空間與雜訊子空間正交，因此上式亦可表示為：

$$\theta_{DOA} = \max \left( a^H(\theta) \left( \sum_{i=1}^d V_i V_i^H \right) a(\theta) \right) \quad (3.3.12)$$

MUSIC 演算法利用 array manifold vector 代表的麥克風之間的相位差關係，以及和 eigenvector 之間的內積尋找正確的聲源角度，代表了 eigenvector 本身即代表了麥克風之間的相位差資訊，因此我們不使用 array manifold vector，而是直接從 eigenvector 估算麥克風之間的相位關係，首先，將最大的 eigenvalue 對應到的 eigenvector  $V_1(\omega_f)$  取出，由  $V_1(\omega_f)$  代表頻率為  $\omega_f$  的聲音訊號在麥克風陣列中的主要分布向量， $V_1(\omega_f)$  可以表示成：

$$V_1(\omega_f) = [V_{11}(\omega_f) \quad V_{12}(\omega_f) \quad \cdots \quad V_{1M}(\omega_f)]^T \quad (3.3.13)$$

而  $V_1(\omega_f)$  包含了聲音訊號在不同的頻率下到達麥克風陣列的到達角度資訊，而最後，我們可以將主要分布矩陣寫為以下的形式：

$$\mathbf{E}_1 = \begin{bmatrix} V_{11}(\omega_1) & V_{11}(\omega_2) & \cdots & V_{11}(\omega_F) \\ V_{12}(\omega_1) & V_{12}(\omega_2) & \cdots & V_{12}(\omega_F) \\ \vdots & \vdots & & \vdots \\ V_{1M}(\omega_1) & V_{1M}(\omega_2) & \cdots & V_{1M}(\omega_F) \end{bmatrix} \quad (3.3.14)$$

在(24)式中，第  $f$  行代表了頻率為  $\omega_f$  的聲音訊號在麥克風陣列中的主要分布向量，接者我們在這裡使用 GCC 估算麥克風與麥克風之間的相位關係，因此我們定義麥克風  $i$  和麥克風  $j$  之間的 eigenstructure-based GCC 為

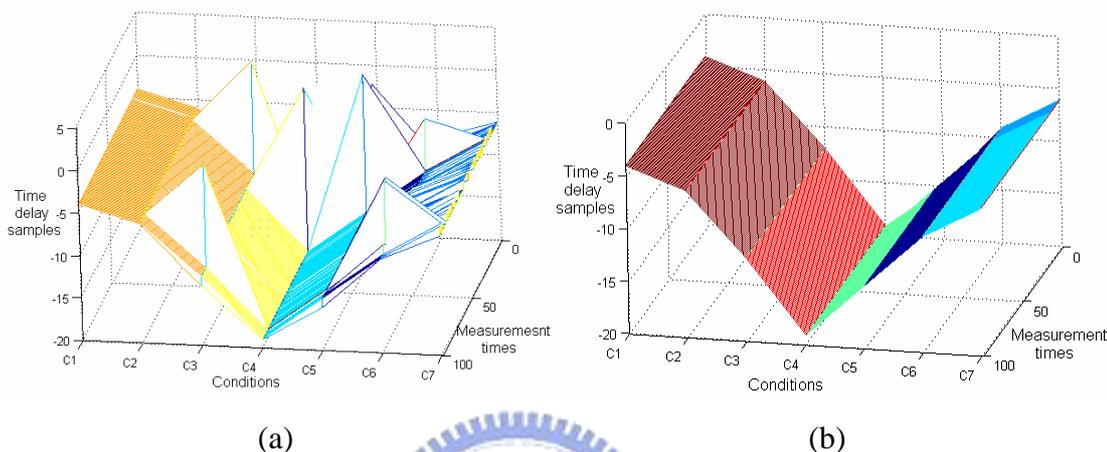
$$R_{x_i x_j}(\tau) = \int_{\omega_1}^{\omega_F} V_{1i}(\omega) V_{1j}^*(\omega) e^{j\omega\tau} d\omega \quad (3.3.15)$$

而麥克風之間的時間延遲量的估算即為：

$$\hat{\tau}_{ES-GCC} = \arg \max_{\tau} R_{x_i x_j}(\tau) \quad (3.3.16)$$

### 3.4 ES-GCC 與 PHAT-GCC 之比較

ES-GCC 主要的效果在於它將聲場中主要的聲音分布計算出來，然後再對主要分布估算時間延遲量，相較於傳統 GCC 直接對訊號估算，ES-GCC 可以降低雜訊分布的影響，所以對於估算時間延遲量而言，ES-GCC 比較能適應不同環境的雜訊影響，下圖二顯示了兩者在時間延遲量估計上的差別。



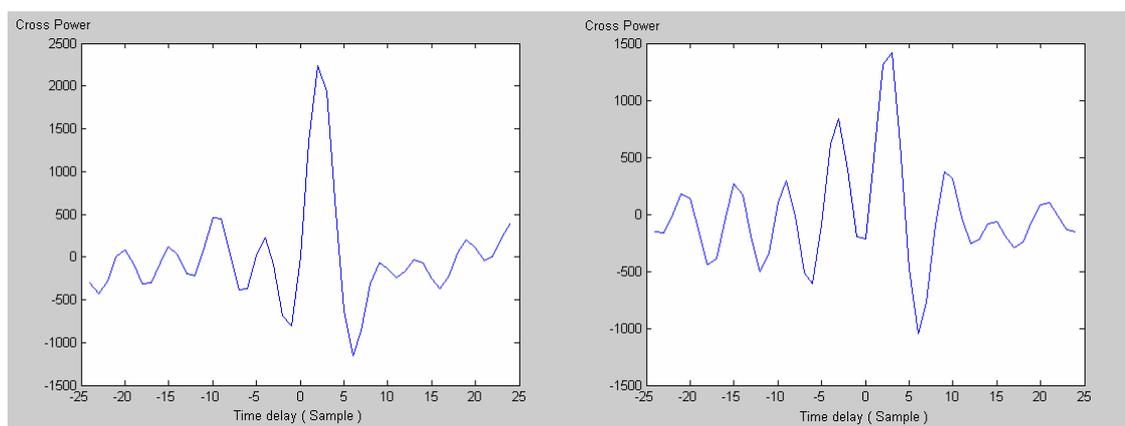
(a)

(b)

圖二、(a)時間延遲估計量(PHAT-GCC)

(b)時間延遲估計量(ES-GCC)

圖二中， $C_i$  代表第  $i$  個麥克風和第一個麥克風之間的比較，從圖中可以很容易觀察出使用 ES-GCC 估算的時間延遲量較為穩定，接下來觀察兩個聲源同時存在的情況，下圖三為此情況時分別使用 ES-GCC 還有 PHAT-GCC 估測時間延遲量的圖



(a)

(b)

圖三、(a) 雙聲源時間延遲估計量(PHAT-GCC)

(b) 雙聲源時間延遲估計量(ES-GCC)

圖三為單純看第二個麥克風和第一個麥克風之間的時間延遲量，圖三(b)中較為明

顯的兩個波峰對應到的時間延遲量，即為兩個聲源在此麥克風對中的時間延遲量，但是觀察圖三(a)並沒有很明顯得突顯出第二個峰值，因為 ES-GCC 可以壓抑雜訊分佈的特性，使得兩個聲源的時間延遲量可以被突顯出來，PHAT-GCC 由於是直接對收到的訊號作處理，所以並沒有這樣的特性，因此兩者在多聲源的情況下之分別可以由此觀察出來。



## 第四章 聲源方位以及聲速估測演算法

### 4.1 簡介

傳統的聲源定位演算法如 MUSIC[24]，必須將聲音傳遞的速度定為已知，雖然關於聲速的數值有理論值可以做為參考，但是實際上，我們並不知道聲音傳遞的速度，K. Yao 等人[25]提出了一套在聲速為未知的情況下依舊能夠估測聲源位置的方法，本論文提出之方法是以此方法為基礎，同時考慮到麥克風陣列的形狀以及聲源距離對估計上的影響，重新推導出了一套聲源方位以及聲速估測演算法，以下將詳細說明及推導此演算法，並且比較兩者之不同。

### 4.2 聲源方位以及聲速估測原理

#### 4.2.1 聲源距離對於方位以及聲速估測的影響

根據[25]提出的方法([25]中第(13)式)，我們可以得到以下的式子：

$$-\frac{(\mathbf{r}_i - \mathbf{r}_1) \cdot (\mathbf{r}_s - \mathbf{r}_1)}{v|\mathbf{r}_s - \mathbf{r}_1|} + \frac{|\mathbf{r}_i - \mathbf{r}_1|^2}{2v|\mathbf{r}_s - \mathbf{r}_1|} - \frac{v(t_i - t_1)^2}{2|\mathbf{r}_s - \mathbf{r}_1|} = (t_i - t_1) \quad (4.2.1)$$

其中， $\mathbf{r}_s$  代表聲源的位置向量， $\mathbf{r}_i = [x_i \ y_i \ z_i]$  代表第  $i$  個麥克風的位置向量，而  $t_i - t_j$  代表第  $i$  個麥克風和第  $j$  個麥克風之間的時間延遲量差(在此我們都將  $j$  設定為第一個麥克風)，而  $v$  代表聲速。接者我們定義以下兩個變數：

$$\hat{\mathbf{r}}_s = \frac{\mathbf{r}_s - \mathbf{r}_1}{|\mathbf{r}_s - \mathbf{r}_1|} \quad \text{以及} \quad \rho_i = \frac{|\mathbf{r}_i - \mathbf{r}_1|}{|\mathbf{r}_s - \mathbf{r}_1|} \quad (4.2.2)$$

$\hat{\mathbf{r}}_s$  代表聲源位置向量的單位向量，而  $\rho_i$  代表聲源距離和麥克風陣列大小之間的比例，而當聲源屬於遠聲場的時候， $\rho_i \ll 1$ ，將(4.2.2)代入(4.2.1)，我們可以得到以下的式子：

$$-(\mathbf{r}_i - \mathbf{r}_1) \frac{\hat{\mathbf{r}}_s}{v} + \frac{|\mathbf{r}_i - \mathbf{r}_1| \rho_i}{v} - \frac{1}{v} \frac{v^2(t_i - t_1)^2}{2|\mathbf{r}_s - \mathbf{r}_1|} \frac{\rho_i}{2} = (t_i - t_1) \quad (4.2.3)$$

(4.2.3)式中， $v(t_i - t_1)$  代表聲源到第  $i$  個麥克風和到第 1 個麥克風之間的距離差，我們定義這樣的距離差為  $d_i$ ，也就是：

$$d_i = v(t_i - t_1) = |\mathbf{r}_s - \mathbf{r}_i| - |\mathbf{r}_s - \mathbf{r}_1| \quad (4.2.4)$$

因此，(4.2.3)可以改寫為:

$$-\frac{(\mathbf{r}_i - \mathbf{r}_1)}{v} \cdot \hat{\mathbf{r}}_s + f_i \frac{\rho_i}{2} = (t_i - t_1) \quad (4.2.5)$$

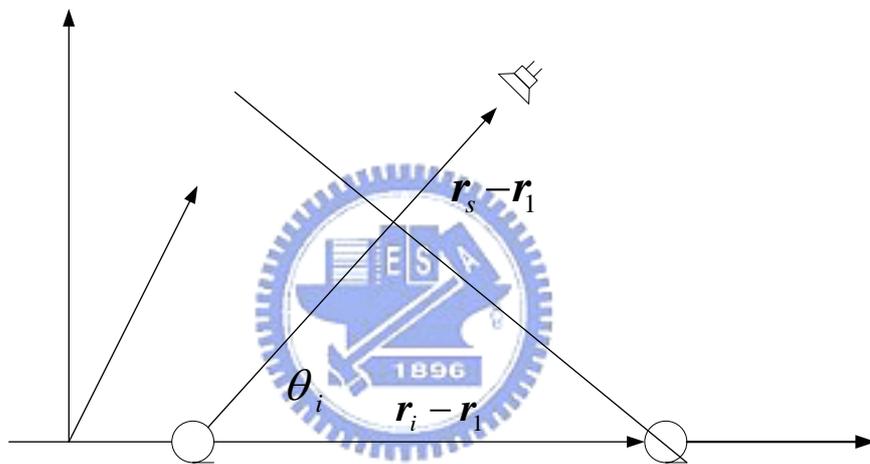
其中

$$f_i = \frac{|\mathbf{r}_i - \mathbf{r}_1|}{v} - \frac{|d_i|}{v} \frac{|d_i|}{|\mathbf{r}_i - \mathbf{r}_1|} \quad (4.2.6)$$

由(4.2.5)式可以發現，當  $\rho_i \ll 1$  的時候，(5)式可以近似為:

$$-\frac{(\mathbf{r}_i - \mathbf{r}_1)}{v} \cdot \hat{\mathbf{r}}_s = (t_i - t_1) \quad (4.2.7)$$

下圖四表示的是一個遠聲場造成的平行波與一組麥克風對之間的關係，這樣的關



圖四、平行波與兩個麥克風之間的幾何關係示意圖

係，可以表示為:

$$\frac{|\mathbf{r}_i - \mathbf{r}_1| \cos(\theta_i)}{v} = t_1 - t_i \quad (4.2.8)$$

而其中  $\cos(\theta_i)$  又可以改寫為:

$$\cos(\theta_i) = \frac{(\mathbf{r}_i - \mathbf{r}_1) \cdot (\mathbf{r}_s - \mathbf{r}_1)}{|\mathbf{r}_i - \mathbf{r}_1| |\mathbf{r}_s - \mathbf{r}_1|} \quad (4.2.9)$$

而(4.2.7)式可以由(4.2.8)、(4.2.9)兩式推導出來，因此我們可以將(7)式視為遠聲場下的關係式，所以我們定義  $\rho_i$  為聲場遠近的比例，而  $f_i$  為近聲場的影響因子，當聲源距離麥克風陣列越來越遠時，原本的關係式(4.2.1)就越接近遠聲場的關係式(4.2.8)。

#### 4.2.2 利用最小平方法求解以及麥克風陣列幾何形狀的影響

由於(4.2.1)式是一個線性的方程式，所以聲源位置的估算可以利用最小平方法解(4.2.1)式而求得，首先我們定義聲源方向單位向量為：

$$\mathbf{w}_s \equiv [w_1 \quad w_2 \quad w_3]^T = \frac{\mathbf{r}_s}{v|\mathbf{r}_s|} = \frac{\hat{\mathbf{r}}_s}{v} \quad (4.2.10)$$

以及定義另外兩個變數為：

$$w_4 = \frac{1}{2v|\mathbf{r}_s - \mathbf{r}_1|}, w_5 = \frac{v}{2|\mathbf{r}_s - \mathbf{r}_1|} \quad (4.2.11)$$

因此 M 個麥克風的線性方程式可以寫成：

$$\mathbf{A}\mathbf{w} = \mathbf{b} \quad (4.2.12)$$

其中

$$\mathbf{A} = \begin{bmatrix} -(\mathbf{r}_2 - \mathbf{r}_1) & |\mathbf{r}_2 - \mathbf{r}_1|^2 & -(t_2 - t_1)^2 \\ -(\mathbf{r}_3 - \mathbf{r}_1) & |\mathbf{r}_3 - \mathbf{r}_1|^2 & -(t_3 - t_1)^2 \\ \vdots & \vdots & \vdots \\ -(\mathbf{r}_M - \mathbf{r}_1) & |\mathbf{r}_M - \mathbf{r}_1|^2 & -(t_M - t_1)^2 \end{bmatrix} \quad (4.2.13)$$

$$\mathbf{b} = \begin{bmatrix} t_2 - t_1 \\ t_3 - t_1 \\ \vdots \\ t_M - t_1 \end{bmatrix} \quad (4.2.14)$$

$$\mathbf{w} = \begin{bmatrix} w_s \\ w_4 \\ w_5 \end{bmatrix} \quad (4.2.15)$$

在 3D 的環境下，需要求的變數有 5 個，因此需要至少 6 個的麥克風才可以求解，解為：

$$\mathbf{w} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \quad (4.2.16)$$

而聲源位置以及聲速的估算如下：

$$\mathbf{r}_s = \frac{w_5}{w_4} \mathbf{r}_1, v = \sqrt{\frac{w_5}{w_4}} \left( \text{or } v = \frac{1}{|\mathbf{w}_s|} \right) \quad (4.2.17)$$

當聲場接近遠聲場模型的時候( $\rho_i \ll 1$ )，由(4.2.7)式可以得到新的方程式為：

$$\mathbf{A}_s \mathbf{w}_s = \mathbf{b} \quad (4.2.18)$$

其中

$$\mathbf{A}_s = \begin{bmatrix} -(\mathbf{r}_2 - \mathbf{r}_1) \\ -(\mathbf{r}_3 - \mathbf{r}_1) \\ \vdots \\ -(\mathbf{r}_M - \mathbf{r}_1) \end{bmatrix} \quad (4.2.19)$$

由於使用遠聲場模型的關係，我們無法求得聲源的座標，只能得到聲源的方向：

$$\hat{\mathbf{r}}_s = \frac{\mathbf{w}_s}{|\mathbf{w}_s|} = \frac{(\mathbf{A}_s^T \mathbf{A}_s)^{-1} \mathbf{A}_s^T \mathbf{b}}{[(\mathbf{A}_s^T \mathbf{A}_s)^{-1} \mathbf{A}_s^T \mathbf{b}]} \quad (4.2.20)$$

而聲速的估算為：

$$v = \frac{1}{|\mathbf{w}_s|} = \frac{1}{[(\mathbf{A}_s^T \mathbf{A}_s)^{-1} \mathbf{A}_s^T \mathbf{b}]} \quad (4.2.21)$$

矩陣  $\mathbf{A}$  和  $\mathbf{A}_s$  必須是 full rank，(4.2.12)式以及(4.2.18)式才有辦法求解，而在 3D 的情況下，從  $\mathbf{A}_s$  可以很容易的觀察出來，麥克風的分佈必須展出一個 3D 的空間(必須是立體的)， $\mathbf{A}_s$  才會是 full rank 的，但是矩陣  $\mathbf{A}$  的情況卻不同，就算麥克風陣列的幾何形狀是立體的，矩陣  $\mathbf{A}$  還是有可能不是 full rank 的，以下將以球面分佈為例去證明球面分佈的麥克風陣列依舊無法使用矩陣  $\mathbf{A}$  求解。根據(4.2.12)式中  $\mathbf{A}$  的定義，首先假設麥克風的向量座標為：

$$\bar{\mathbf{r}}_1 = [R \sin \phi_1 \cos \theta_1 \quad R \sin \phi_1 \sin \theta_1 \quad R \cos \phi_1] \quad (4.2.22)$$

$$\bar{\mathbf{r}}_i = [R \sin \phi_i \cos \theta_i \quad R \sin \phi_i \sin \theta_i \quad R \cos \phi_i] \quad (4.2.23)$$

利用(4.2.22)和(4.2.23)式，我們可以定義  $-(\bar{\mathbf{r}}_i - \bar{\mathbf{r}}_1)$  為：

$$-(\bar{\mathbf{r}}_i - \bar{\mathbf{r}}_1) = [-R(\sin \phi_i \cos \theta_i - \sin \phi_1 \cos \theta_1) \quad -R(\sin \phi_i \sin \theta_i - \sin \phi_1 \sin \theta_1) \quad -R(\cos \phi_i - \cos \phi_1)] \quad (4.2.24)$$

而  $|\bar{\mathbf{r}}_i - \bar{\mathbf{r}}_1|^2$  則可由下式推導出來：

$$\begin{aligned} |\bar{\mathbf{r}}_i - \bar{\mathbf{r}}_1|^2 &= R^2[(\sin \phi_i \cos \theta_i - \sin \phi_1 \cos \theta_1)^2 + (\sin \phi_i \sin \theta_i - \sin \phi_1 \sin \theta_1)^2 + (\cos \phi_i - \cos \phi_1)^2] \\ &= 2R^2 - 2R^2 \sin \phi_i \cos \theta_i \sin \phi_1 \cos \theta_1 - 2R^2 \sin \phi_i \sin \theta_i \sin \phi_1 \sin \theta_1 - 2R^2 \cos \phi_i \cos \phi_1 \end{aligned} \quad (4.2.25)$$

而觀察(4.2.24)式和(4.2.25)式後可以發現，(4.2.25)式等於(4.2.24)式中 3 項的線性

組合，所以造成矩陣  $A$  無法 full rank，因此無法使用最小平方法求解。

由以上的證明可以了解到麥克風陣列的幾何形狀對於矩陣  $A$  有較大的影響，因此本論文主要是使用遠聲場的模型建立的方程式求出聲源方向以及聲速，好處是麥克風陣列幾何形狀的影響較小，聲速估測較為穩定，而壞處則是無法得到聲源的座標，只能得到聲源的方向，在下一節內將會以模擬以及實驗的結果說明兩者的差別。



### 4.2.3 估測誤差分析

本論文提出之聲源定位演算法主要以遠聲場模型為主，而在使用遠聲場模型對聲源作定位的時候，主要的誤差來源分為兩個部份，第一是在使用 ES-GCC 估算時間延遲量的時候產生的估測誤差，第二是聲源距離影響遠聲場模型的準確性，因此可以將(4.2.5)式改寫為:

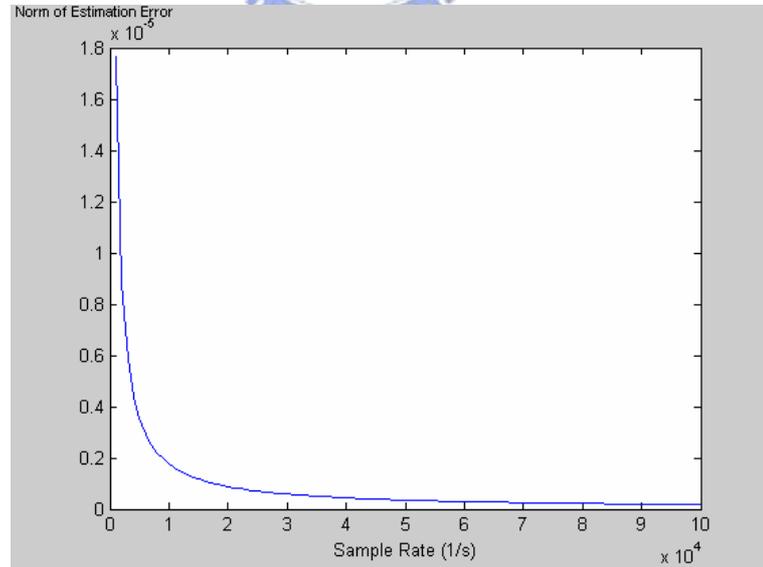
$$-\frac{(\mathbf{r}_i - \mathbf{r}_1)}{v} \cdot (\hat{\mathbf{r}}_s + \hat{\mathbf{r}}_{error}) = (t_i - t_1)_{real} + (t_i - t_1)_{error} - f_i \frac{\rho_i}{2} \quad (4.2.26)$$

其中  $(t_i - t_1)_{error}$  代表時間延遲量估計時造成的誤差項，而  $\hat{\mathbf{r}}_{error}$  代表所有誤差產生的一個誤差向量，此誤差向量就是主要影響到聲源方位估測的部份，而誤差向量的大小會決定影響估測結果的多寡。

首先分析 ES-GCC 部份之誤差，由於麥克風收到的聲音資料受限於 sample rate，因此估計時間延遲量的時候，時間延遲量的解析度就取決於 sample rate 的大小，因此時間延遲量部分的誤差可以表示為:

$$\max \left\{ (t_i - t_1)_{error} \right\} = \frac{1}{2f_s} \quad (4.2.27)$$

因此時間延遲量估計部分的誤差和 sample rate 的大小成反比，由下圖五可以看出



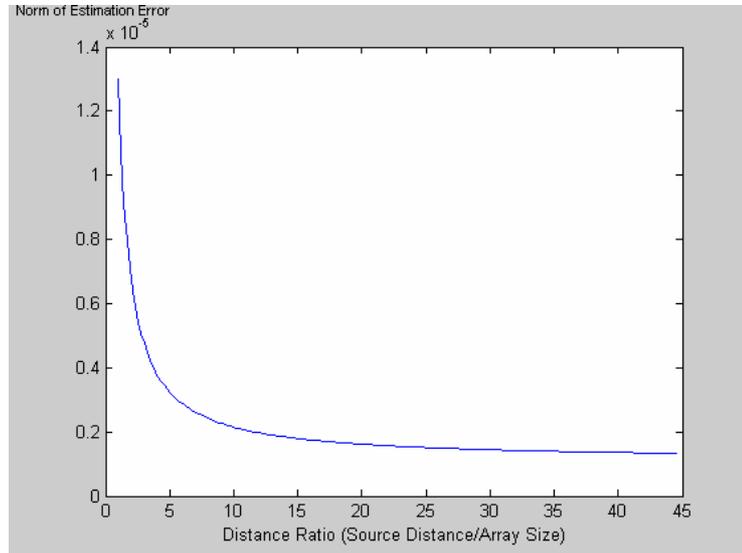
圖五、Sample rate 對誤差的影響

在上圖四中我們以誤差向量的範數(Norm)作為誤差大小的依據：

$$\text{Norm of Estimation Error} \equiv \text{Norm}(\hat{\mathbf{r}}_{error}) = \sqrt{x_{error}^2 + y_{error}^2 + z_{error}^2} \quad (4.2.28)$$

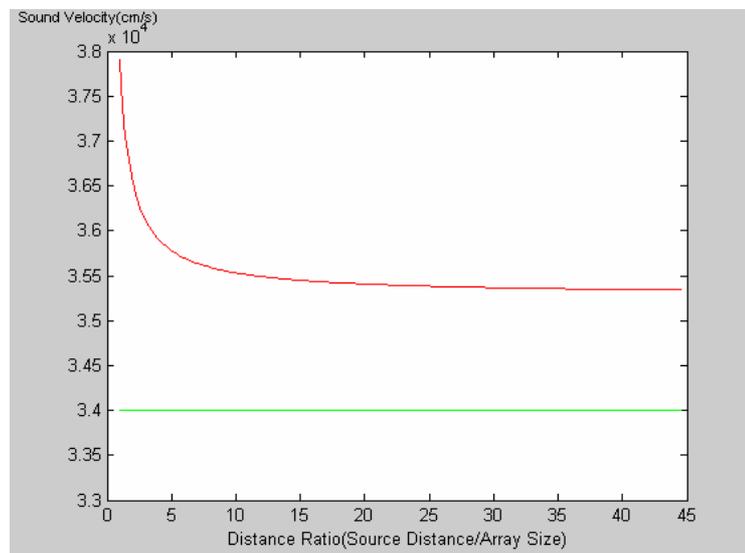
而從圖中可以看出，隨著 sample rate 的提升，估測誤差會因此而降低。

接下來將分析聲源距離對於誤差的影響，根據(4.2.26)式中距離的影響，可以模擬出下圖六，圖中的橫軸代表聲源距離和麥克風陣列大小的比例，將麥克風陣



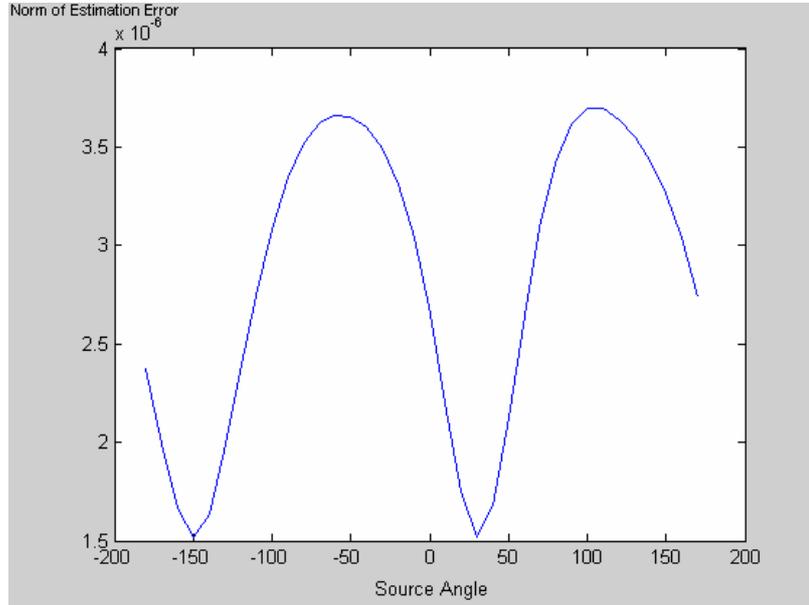
圖六、聲源距離對誤差向量大小的影響

列中最遠的兩個麥克風之間的距離視為麥克風陣列的大小(Array Size)，由圖中可以發現，聲源距離麥克風陣列五倍麥克風陣列大小以內的時候，估測誤差較大而且變化較劇烈，而五倍以外的誤差已經穩定並且收斂在一個較小的數值，因為當聲源距離麥克風陣列越近，遠聲場的模型就越不適用，因此會有較大的誤差產生，而這樣的誤差對於聲速估測的影響如下圖七，圖中紅線代表估測的結果，而綠線則代表聲速理論值(34000 cm/s)。



圖七、聲源距離對聲速估測的影響

而根據  $f_i \frac{\rho_i}{2}$  在第(4.2.2)式以及第(4.2.6)式內的定義我們可以發現聲源所在的角度也會影響誤差的大小，對距離原點相同距離的聲源而言，聲源所在的角度會影響聲源和第一個麥克風的距離，也就是  $|r_s - r_1|$  的大小，因此會影響到誤差的大小，而聲源所在角度和誤差之間的關係可以由下圖八顯示，從圖中誤差值最小的



圖八、聲源角度與誤差大小之關係

兩個角度正好是在第一個麥克風的方向上，原因是因為再演算法中我們主要是使用第一個麥克風做為參考的麥克風，也就是在誤差項中  $|r_s - r_1|$  的大小會影響誤差的大小，因此第一個麥克風的相對位置也會決定誤差的大小，而同樣的我們也可以選擇其他麥克風做為參考的麥克風，而此時最小的誤差就會出現在那個麥克風的角度。

相較於使用遠聲場模型所造成的影響，直接使用 K. Yao 等人提出之(4.2.1)式估測聲源的位置似乎不會有聲源遠近所造成的誤差，但是 sample rate 的影響依舊存在，並且因為矩陣  $\mathbf{A}$  的特性使得 sample rate 同時影響矩陣  $\mathbf{A}$ ，因此也在使用最小評方法求解時產生新的誤差影響，將誤差的影響加入考慮後，(4.2.1)式可以改寫為：

$$-\frac{(\mathbf{r}_i - \mathbf{r}_1) \cdot (\mathbf{r}_s - \mathbf{r}_1)}{v|\mathbf{r}_s - \mathbf{r}_1|} + \frac{|\mathbf{r}_i - \mathbf{r}_1|^2}{2v|\mathbf{r}_s - \mathbf{r}_1|} - \frac{v((t_i - t_1)_{real} + (t_i - t_1)_{error})^2}{2|\mathbf{r}_s - \mathbf{r}_1|} = (t_i - t_1)_{real} + (t_i - t_1)_{error} \quad (4.2.29)$$

$(t_i - t_1)_{error}$  代表了 sample rate 造成的時間延遲量估測誤差，而下圖九為根據(4.2.29)

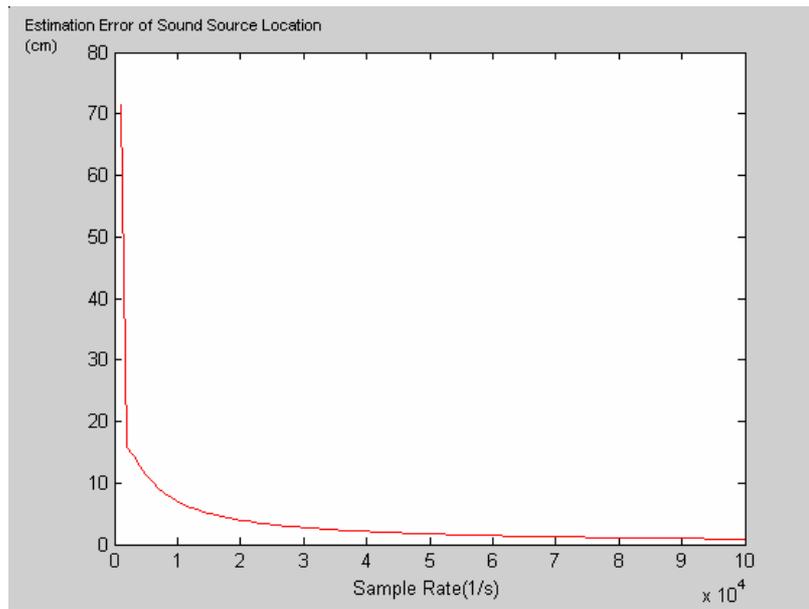


圖 九、Sample rate 對位置估測的影響

式模擬 sample rate 對於位置估測誤差的影響，此時聲源距離陣列 80 公分，sample rate 與誤差的關係和遠聲場模型有一樣的特性，本論文使用的 sample rate 為 16kHz，由圖七中可以觀察出誤差大約在 10 公分以內，但是當聲源距離改變的時候，誤差和聲源距離的關係可以由下圖十模擬出來，從圖中可以看到當聲源距離

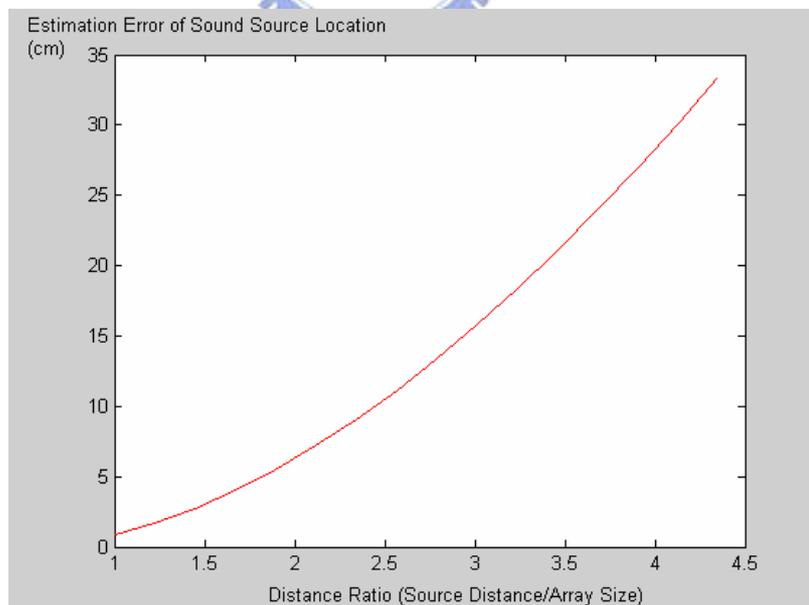


圖 十、聲源距離與位置估測誤差關係

麥克風陣列兩倍 Array Size 以內的時候，誤差還可以在 5 公分以內，但是聲源距離越遠，位置估測的誤差就不斷的增加，這樣的現象導致此演算法在實際的情況

下將無法使用，因此本論文主要使用遠聲場模型估測聲源方位。

根據演算法中矩陣  $\mathbf{A}_s$  的描述，理論上在 3D 的空間上，我們只需要 4 個麥克風就可以計算聲源的方位，而麥克風數量的提升對於估計上的影響可以由下圖十

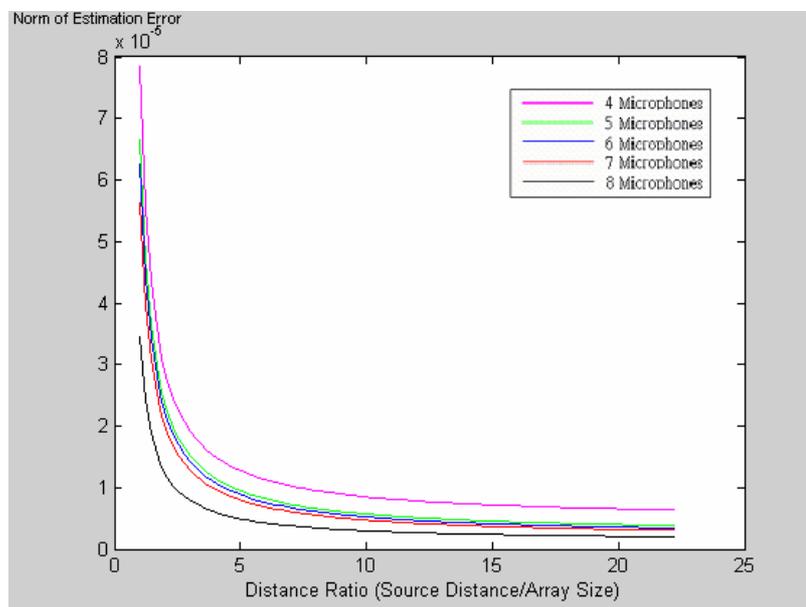


圖 十一、麥克風數量對誤差的影響

一中觀察出來，隨者麥克風數量的提升，估計誤差的量下降的速度有變快，並且最後收斂到較小的數值，因此可以知道麥克風數量變多有助於誤差的收斂。

### 4.3 聲源方位估測結果

接下來將說明實驗的結果，實驗中使用的麥克風數量為 8 顆麥克風，sample rate 則是 16 kHz，而實驗的環境為吵雜的正常環境，麥克風陣列與聲源位置的示

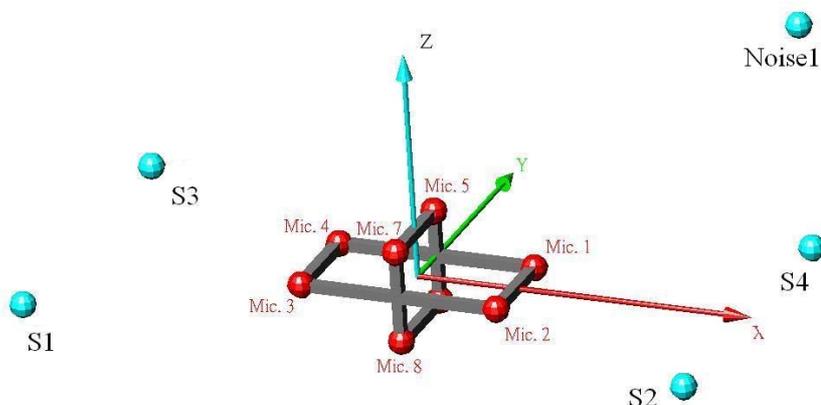


圖 十二、麥克風陣列與聲源相對位置示意圖

意圖如上圖十二，聲源距離麥克風陣列 240 公分，而聲源方位的辨識結果如表一

Experimental Conditions			Experimental Results
Source	SNR (dB)	Correct Angle ( $\theta$ $\phi$ )	Mean
S1	14.78	(-153°, -6°)	(-152.74°, -6.15°)
S2	15.51	(-27°, -3°)	(-28.61°, -3.05°)
S3	16.35	(207°, 0°)	(207.89°, 0°)
S4	16.21	(27°, 6°)	(25.2°, 6.07°)

表 一、聲源方位估算結果

所示，仰角以及水平角的誤差範圍大約是兩度以內。

## 第五章 多聲源方位偵測與聲源數量估算演算法

### 5.1 簡介

傳統的多聲源方位偵測在前面的介紹中有提到，像是 Eigenspace Method 可以做得到，但是必須在聲源數量為已知的情況下才可以估算，但是在實際的情況下，聲源數量是無法在估算前就能獲得的資訊，因此本論文提出的多聲源方位偵測演算法可以在聲源數量為未知的情況下分離不同聲源在麥克風陣列中的時間延遲量，並且估測分別的方位，主要的方法是藉由不同的時間延遲量之排列組合計算出的聲速的正確性，判斷出正確的排列組合，然後經由時間的累積，透過 Adaptive K-means++ 演算法的遞迴計算，最後就可以獲得聲源數量的資訊，以下將分別對多聲源方位估測演算法以及 Adaptive K-means++ 演算法做詳細的介紹。

### 5.2 多聲源方位估測演算法

根據第四章第二節介紹的聲源方位估算的方法我們可以了解到，正確的一組時間延遲量組合，可以推算出一個聲源正確的方位以及合理的聲速，而多聲源同時存在時，如果能夠透過時間延遲量估算的部份取出每個聲源正確的時間延遲量組合，就可以估算每個聲源的到達角度，對於兩個麥克風之間的時間延遲量來說，透過相關性的大小我們可以決定要取出幾種不同的時間延遲量，以下圖十三

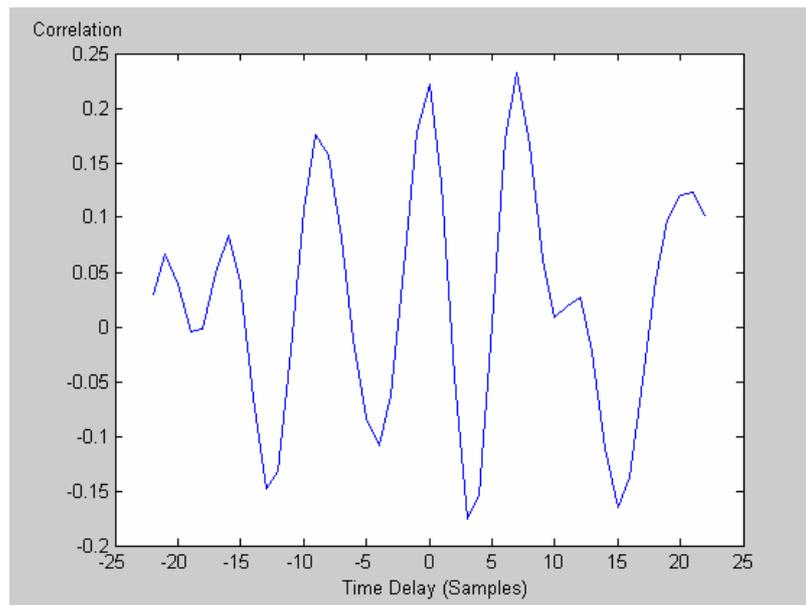


圖 十三、多聲源 ES-GCC 結果圖

為例，此圖為兩個麥克風在同時有四個聲音存在的時候使用 ES-GCC 計算出的結果，縱軸代表相關性，橫軸代表時間延遲量，而其相對應到的相關性越高，代表麥克風接收到的聲音訊號中越有可能有訊號是這樣的時間延遲量，因此首先假設圖十三中相關性的最大值為有聲源存在的相關性數值，然後設定此數值的 0.9 倍

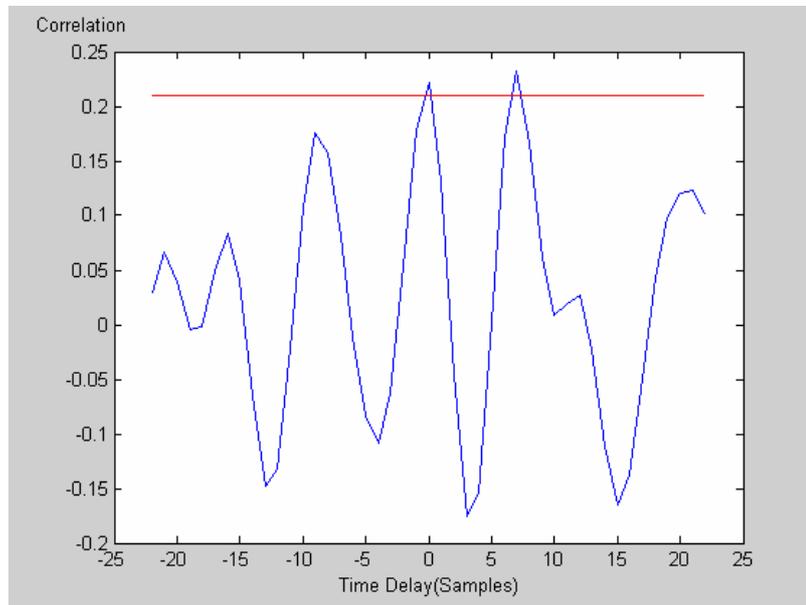


圖 十四、與相關性最大值的 0.9 倍比較後的 ES-GCC 結果圖

為有聲源存在的基本相關性數值，因此如上圖十四所示，在這兩個麥克風之間，我們可以取出兩個不同的時間延遲量，分別是 0 個 sample 以及 7 個 sample，這樣的結果並不代表空間中只有兩個聲源，因為兩個不同的聲源在一組麥克風對中

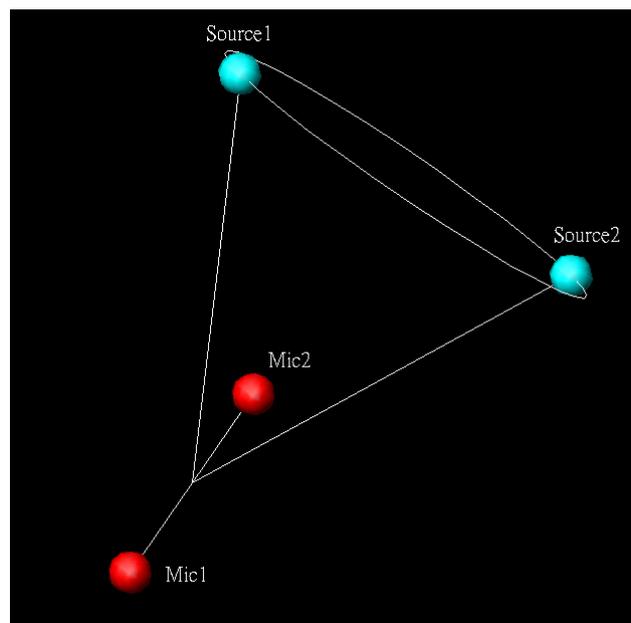


圖 十五、空間中雙聲源與麥克風對之位置關係圖

可以有相同的時間延遲量，由上圖十五可以說明這個現象，如果空間中兩個聲源的位置對於一組麥克風對來說是在同一個圓錐面上，那這兩個聲源的訊號在此麥克風對中表現的時間延遲量將相同，因此單就以上收集到的兩個時間延遲量並不能直接說明聲源的數量。

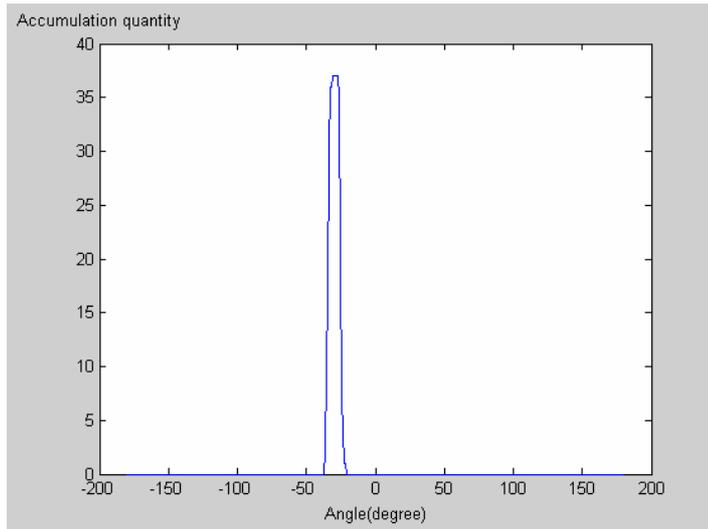
接者我們對每個麥克風和第一個麥克風之間進行和上一段相同的時間延遲量估算，而得到的時間延遲量組合如下表二，我們獲得了一些可能是代表聲源的

	Time Delay(sample)		
Mic1-Mic2	0	7	
Mic1-Mic3	5	2	1
Mic1-Mic4	1		
Mic1-Mic5	3		
Mic1-Mic6	3	-1	4
Mic1-Mic7	2	1	
Mic1-Mic8	-1	-6	

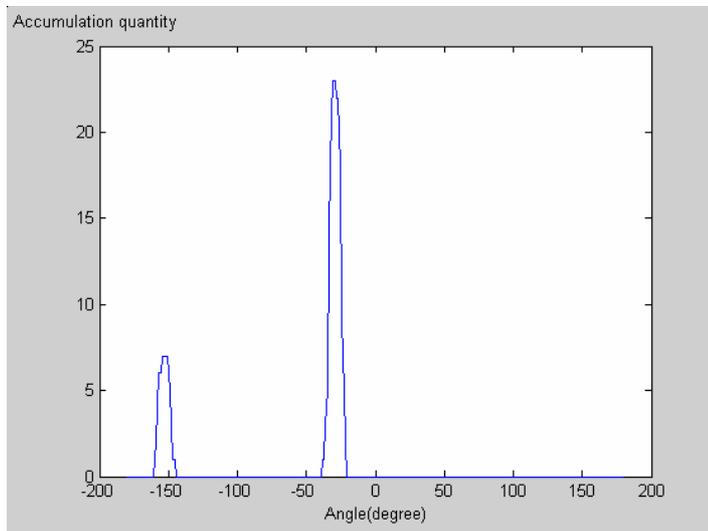
表 二、時間延遲量組合

時間延遲量，而根據第四章介紹的演算法，我們必須知道正確的時間延遲量組合才能估算出正確的聲源方位，而正確的一組時間延遲量組合也可以計算出合理的聲速，我們可以藉由聲速估算的正確性，判斷是否為正確的時間延遲量組合，因此我們將表二中所有的排列組合取出，表二的情況一共會有 72 種不同的時間延遲量組合，而接下來是估算所有組合的聲速，然後將聲速估算合理的时间延遲量組合紀錄下來，全部測試完畢之後，我們就獲得組合後的時間延遲量，而聲速估算正確的時間延遲量組合對應到的方向就是空間中可能的聲源方向。

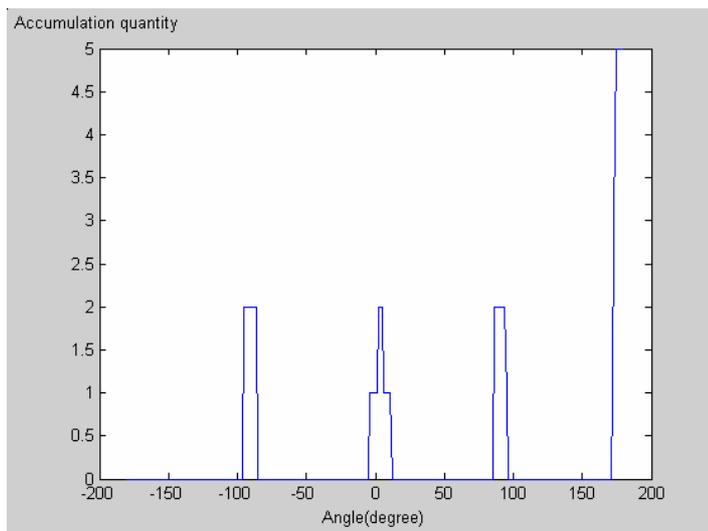
而隨著時間的累積，不同的聲源數量產生的角度分布圖如下圖十六，從圖中可以很清楚的看出不同的聲源數量會產生不同的角度分布，而接下來就需要分類的方法將資料自動分類以及計算出一共有幾個聲源。



(a)



(b)



(c)

圖 十六、估算聲源角度分布(a)單聲源(b)雙聲源(c)四聲源

## 5.3 Adaptive K-means++演算法

從圖十四可以看出，資料分布已經有分群的特性，而此時我們需要的是一個自動計算群數並且對資料做分類的自動分類演算法，在這裡我們使用的是 Adaptive K-means++演算法，此演算法是改良傳統的 K-means++而產生的，傳統 K-means 演算法的特性是當我們決定了要分的群數後(K)，K-means 演算法會自動的將資料分為 K 群，並且計算出每一群的群中心，而改良後的 Adaptive K-means++演算法則是依照資料分群的特性，自動計算出應該要將資料分為幾群，然後再對資料做分群的動作

### 5.3.1 K-means++演算法

K-means++演算法是由 D. Arthur 與 S. Vassilvitskii 提出的[26]，是改良自傳統 K-means 演算法，傳統的 K-means 演算法可以由以下的步驟說明其運作的流程：

1. 決定要分的群數(K)，並且隨機的選取每一群出使的群中心數值。
2. 將數據資料分為 K 群，沒一群內的資料是由數據資料中距離此群中心最近的資料組成。
3. 重新計算分群後每一群的群中心，計算方式為： $C_i = \frac{1}{N_i} \sum c_i$ ，其中  $C_i$  代表第 i 群新的群中心，代表  $N_i$  第 i 群內資料的數目，而  $c_i$  代表第 i 群內的所有數據資料。
4. 重複第二以及第三步驟，直到群中心不再移動為止。

經由以上的步驟，K-means 演算法的確可以依照資料分散的特性將資料分為 K 群，但是在分群的過程中，群中心的初始值對於分群的結果會有很大的影響，因為分群的動作是由群中心的初始值開始往資料延伸的，而由於 K-means 演算法中每一群群中心的初始值是隨機產生的，因此對同一筆數據資料而言，每次使用 K-means 演算法分群的效果可能有很大的差別，可能因此而產生錯誤的分群結果，不好的初始值也會大大的降低分群的速度，因此需要比較有效的初始值選取方式。

K-means++演算法在群中心初始值的選取上提供了一個有效的方法，初始值選取的步驟如下：

1. 在數據資料中隨機選取一個資料作為第一群中心的初始值( $C_1$ )。
2. 令下一群的群中心初始值  $C_i = x'$ ，而  $x'$  的選取方式為  $\max_x \left( \frac{D(x)^2}{\sum D(x)^2} \right)$ ，其中  $D(x)$  代表距離數據資料  $x$  和距離此數據資料最近的群中心的距離， $x$  代表數據資料中所有的資料。
3. 重複步驟 2，直到已經選出  $K$  個初始的群中心。

依照以上步驟選取的群中心的特性是，可以很均勻的分散在數據資料中，初始值不會特別的集中在某一群內，這樣可以有效的避免分群的錯誤產生，並且可以加快分群的速度。

雖然 K-means++ 演算法已經提出了一套出有效的初始值選取方法，但是第一群中心的初始值依舊是隨機產生的，並且在選取初始值之前，我們依然要預先決定需要分的群數( $K$ )，因此本論文在這兩點上做了改良，並將其衍生為可以估算聲源數量的 Adaptive K-means++ 演算法。

### 5.3.2 Adaptive K-means++ 演算法

由於聲源數量在本論文的情況下是未知的，我們無法直接定出數據資料要分為幾群，使得原始的 K-means++ 演算法無法直接的使用，因此我們改良了 K-means++ 演算法，先由只分一群開始，藉由分群後群內的分散程度判斷是否需要增加分群數，直到分群的結果滿意為止，分群的流程可以由下圖十七說明：

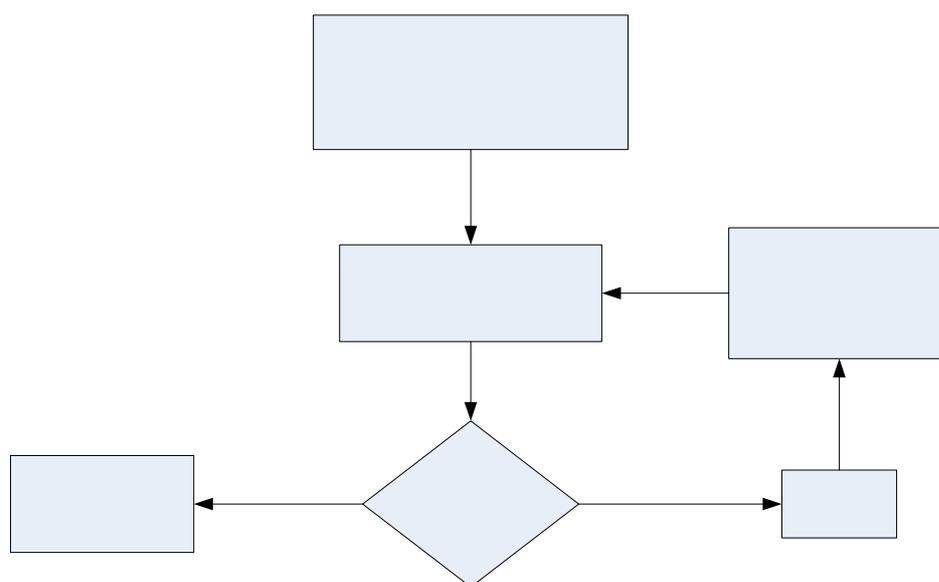


圖 十七、Adaptive K-means++ 演算法流程圖

根據 4.2.3 中分析的聲源方位估算誤差，我們將誤差向量的影響轉換成角度的誤差後可以獲得下圖十八，圖中藍線代表角度誤差與聲源距離的關係，而紅線則是

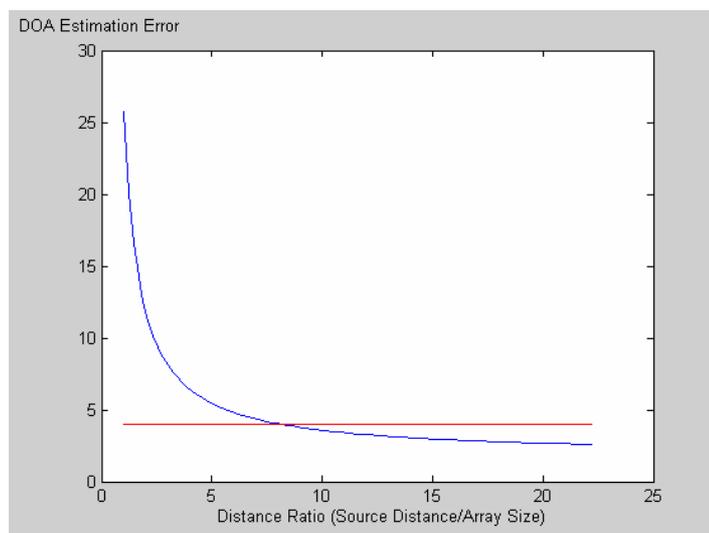


圖 十八、聲源距離與角度估算誤差關係圖

誤差為 4 度的位置，從圖中我們可以觀察出當聲源離麥克風陣列一定的距離後，角度的誤差將收斂至 4 度以下，因此設定角度估算的結果每一群內的角度範圍為 4 度以內，也就是變異數要小於 16，分群結果必須符合此條件才算是分群完成，因此經由 5.2 提出的方法計算出的數據資料先由只分一群開始，然後依據群內的分散程度決定是否要增加群數，直到每一分群內的角度變異數小於 16 則完成分群，最後的群數即代表聲源數量，而群中心即為每個聲源的方位。

## 5.4 多聲源方位偵測與聲源數量估測結果

實驗是在正常吵雜的環境中進行，使用 8 個麥克風構成的麥克風陣列，麥克風陣列如下圖十九所示，首先是單聲源的情況，下表三代表單聲源的情況經由

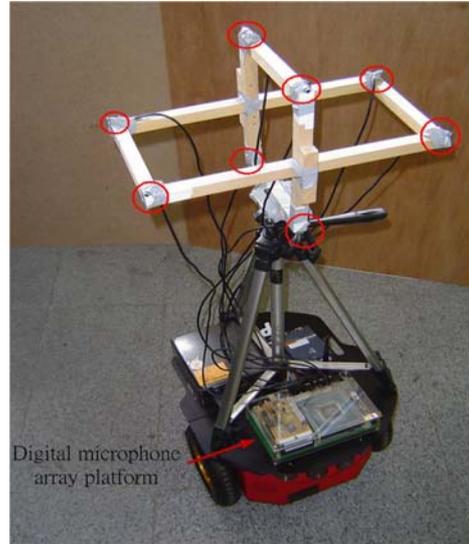


圖 十九、麥克風陣列

Experimental Conditions			Experimental Results	Experimental Results
Source	SNR (dB)	Correct Angle ( $\theta$ $\phi$ )	Mean	Source Number Estimation Results
S1	14.78	(-153°, -6°)	(-152.74°, -6.15°)	1
S2	15.51	(-27°, -3°)	(-28.61°, -3.05°)	1
S3	16.35	(207°, 0°)	(207.89°, 0°)	1
S4	16.21	(27°, 6°)	(25.2°, 6.07°)	1

表 三、單聲源方位與聲源數量估算結果

此方法估算後的情況，單聲源估算的情況較為單純，而接下來觀察同時存在兩個聲源的時候，聲源方位估算與聲源數量估算的結果，結果如下表四，經由以上提

Experimental Conditions			Experimental Results	Experimental Results
Source	SNR (dB)	Correct Angle ( $\theta$ $\phi$ )	Mean	Source Number Estimation Results
S1	13.38	(-153°, -6°)	(-152.74°, -6.15°)	2
S2		(-27°, -3°)	(-28.61°, -3.05°)	
S2	13.53	(-27°, -3°)	(-28.61°, -3.05°)	2
S3		(207°, 0°)	(207.25°, 0°)	
S3	14.24	(207°, 0°)	(206.56°, -2.01°)	2
S4		(27°, 6°)	(26.57°, 6.07°)	

表 四、雙聲源方位與聲源數量估算結果

出之估算方法，同時存在雙聲源的時候依舊可以找出各別的方位以及估算出聲源數量，接者觀察同時有四個聲源存在於空間中的時候估算的結果，如下表五所示，

Experimental Conditions			Experimental Results	Experimental Results
Source	SNR (dB)	Correct Angle ( $\theta \phi$ )	Mean	Source Number Estimation Results
S1	15.38	$(-0.25^\circ, 1.2^\circ)$	$(1.35^\circ, 0.32^\circ)$	4
S2		$(88.45^\circ, 1^\circ)$	$(90.13^\circ, 0.45^\circ)$	
S3		$(-92.6^\circ, 0.83^\circ)$	$(-90.36^\circ, 0.36^\circ)$	
S4		$(178.65^\circ, 1.7^\circ)$	$(179.65^\circ, 0.12^\circ)$	

表 五、同時四聲源方位與聲源數量估測結果

由以上結果可以驗證在同時有多聲源存在的情況下，此演算法可以對個別聲源角度做估測的動作，並且可以估算出聲源數量。



## 第六章 未來展望

本論文中我們提出了一套利用分散式的麥克風陣列在只有麥克風座標以及麥克風收集到的聲音資訊下對多聲源方位以及聲源數量估測的方法，透過這樣的方法我們可以處理同時存在多個聲源的情況，並且估算出實際空間中存在的聲源數量，由於麥克風可以是分散式的不需要有特定的形狀，所以更能符合各種應用，也因為此聲源方位偵測的方法不需要知道聲源數量，因此更能適用實際的空間狀態。

目前使用的麥克風陣列有 8 個麥克風，根據第四章的分析可以知道，越多麥克風對於誤差的降低是有幫助的，而誤差的降低不僅是代表角度估測正確性的提升同時意味著聲源的數量可以更多，未來可以應用在人數的估計以及輔助保全攝影機監控雜亂的會場方面，另外可以應用在人機介面上，做到多人同時與機器人互動。



## Reference:

- [1]. *P. R. Roth*, "Effective measurements using digital signal analysis," *IEEE Spectrum*, vol. 8, pp. 62-70, Apr. 1971.
- [2]. *G. C. Carter, A. H. Nuttall, and P. G. Cable*, "The smoothed coherence transform(SCOT)," *Naval Underwater Systems Center, New London Lab., New London Lab., New London, CT, Tech. Memo TC-159-72*, Aug. 8, 1972.
- [3]. *G. C. Carter, A. H. Nuttall, and P. G. Cable*, "The smoothed coherence transform," *Proc. IEEE (Lett.)*, vol. 61, pp. 1497-1498, Oct. 1973.
- [4]. *C. H. Knapp and G. C. Carter*, "The generalized correlation method for estimation of time delay," *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-24(4):320-327, Aug. 1976.
- [5]. *B. Champagne, S. Bedard, and A. Stephenne*, "Performance of Time-Delay Estimation in the Presence of Room Reverberation," *IEEE Trans. of Speech and Audio Processing*, vol. 4, no. 2, March 1996.
- [6]. *A. Stephenne, and B. Champagne*, "Cepstral prefiltering for time delay estimation in reverberant environments," *ICASSP-95.*, vol. 5, May 1995.
- [7]. *M. S. Brandstein, H. F. Silverman*, "A Robust Method for Speech Signal Time-Delay Estimation in Reverberant Rooms," *ICASSP-97*, vol. 1, April 1997.
- [8]. *M. Omologo and P. Svaizer*, "Use of the Cross-power-Spectrum Phase in Acoustic Event Location," *IEEE Trans. of Speech and Audio Processing*, vol. 5, no. 3, May 1997.
- [9]. *Hu, J., Su, T.M., Cheng, C.C., Liu, W.H., and Wu, T.I*, "A self-calibrated speaker tracking system using both audio and video data", *IEEE Conference on Control Applications*, Sep, 2002.
- [10]. *Hu, J., Cheng, C.C., Liu, W.H., Su, T.M.*, "A Speaker Tracking System with Distance Estimation Using Microphone Array", *IEEE/ASME International Conference on Advanced Manufacturing Technologies and Education*, Aug, 2002.
- [11]. *M. Xinyu and C. L. Nikias*, "Joint Estimation of Time Delay and Frequency Delay in Impulsive Noise Using Fractional Lower Order Statistics," *IEEE Trans. of Signal Processing*, vol. 44, no. 11, Nov. 1996.

- [12]. *P. G. Georgiou and P. Tsakalides*, “Alpha-Stable Modeling of Noise and Robust Time-Delay Estimation in the Presence of Impulsive Noise,” *IEEE Trans. of Multimedia*, vol. 1, no. 3 Sep. 1999.
- [13]. *C.L. Nikas and M. Shao*, *Signal Processing with Alpha-Stable Distributions and Applications*. New York: Wiley, 1995.
- [14]. *M. Wax, T. J. Shan, and T. Kailath*. “Spatio-temporal spectral analysis by eigenstructure methods,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol ASSP-32, pp. 817–827, Aug 1984.
- [15]. *H. Wang and M. Kaveh*, “Coherent signal subspace processing for detection and estimation of angle of arrival of multiple wideband sources,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol ASSP-33, pp. 823–831, Aug. 1985.
- [16]. *G. Bienvenu*, “Eigensystem properties of the sampled space correlation matrix,” *Proc. IEEE, ICASSP, Boston MA, 1983*, pp. 332–335.
- [17]. *K. M. Buckley and L. J. Griffiths*, “Eigenstructure based broadband source location estimation,” in *Proc. IEEE ICASSP, Tokyo, Japan, 1986*, pp. 1869–1872.
- [18]. *M. A. Doron, A. J. Weiss, and H. Messer*, “Maximum likelihood direction finding of wideband sources,” *IEEE Trans. Signal Processing*, vol. 41, pp. 411–414, Jan 1993.
- [19]. *Y. Bresler and A. Macovski*, “Exact maximum likelihood parameter estimation of superimposed exponential signals in noise,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 1081-1089, Oct. 1986.
- [20]. *M. Agarwal and S. Prasad*, “DOA estimation of wideband sources using a harmonic source model and uniform linear array,” *IEEE Trans. Signal Processing*, vol. 47, pp. 619-629, Mar. 1999.
- [21]. *H. Messer*, “The potential performance gain in using spectral information in passive detection/localization of wideband sources,” *IEEE Trans. Signal Processing*, vol. 43, pp. 2964-2974, Dec. 1995.
- [22]. *M. Agrawal and S. Prasad*, “Broadband doa estimation using spatial-only modeling of array data,” *IEEE Trans. Signal Processing*, vol. 48, pp. 663-670, Mar. 2000.
- [23]. *J.-H Lee, Y.-M Chen, and C.-C Yeh*, “A covariance approximation method for near-field direction finding using a uniform linear array,” *IEEE Trans. Signal*

*Processing, vol. 43, pp. 1293-1298, May 1995.*

[24]. *R. O. Schmidt*, "Multiple Emitter Location and Signal Parameter Estimation", *IEEE Trans. Antennas and Propag.*, vol. AP-34, no. 3, pp.276-280, March 1986

[25]. *K. Yao, R. E. Hudson, C. W. Reed, D. Chen, and F. Lorenzelli*, "Blind beamforming on a randomly distributed sensor array system," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 1555-1567, Oct. 1998

[26]. *D. Arthur, S. Vassilvitskii*: "k-means++ The Advantages of Careful Seeding" 2007 Symposium on Discrete Algorithms (SODA).

