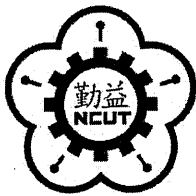


DT  
448.6  
2582  
100  
225320



國立勤益科技大學  
電子工程系研究所

碩士論文

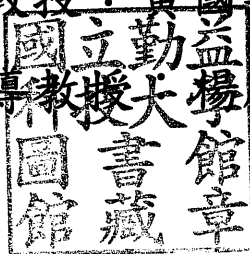
嵌入式自動語音辨識系統應用於智慧型機器人

**The ASR Technique Based on Embedded System  
for Intelligent Robot**

研究生：朱銘傑

指導教授：黃國興 博士

共同指導教授：楊勝智 博士



中華民國 一 百 年 七 月

國立勤益科技大學圖書館



225320

嵌入式自動語音辨識系統應用於智慧型機器人

**The ASR Technique Based on Embedded System  
for Intelligent Robot**

研究生：朱銘傑

指導教授：黃國興 博士

共同指導教授：楊勝智 博士

國立勤益科技大學

電子工程系研究所

碩士論文

Submitted in partial fulfillment of the requirements  
For the degree of  
Master of Engineering  
In  
Institute of Electronic Engineering  
from the National Chin-Yi University of Technology  
July 2011  
Taiping, Taichung, Taiwan, Republic of China

中華民國 一 百 年 七 月

# 國家圖書館 博碩士論文電子檔案上網授權書

本授權書所授權之論文為授權人在國立勤益科技大學電子工程系 99 學年度第 2 學期取得碩士學位之論文。

論文題目：嵌入式自動語音辨識系統應用於智慧型機器人  
指導教授：黃國興, 楊勝智

茲同意將授權人擁有著作權之上列論文全文（含摘要），非專屬、無償授權國家圖書館，不限地域、時間與次數，以微縮、光碟或其他各種數位化方式將上列論文重製，並得將數位化之上列論文及論文電子檔以上載網路方式，提供讀者基於個人非營利性質之線上檢索、閱覽、下載或列印。

上列論文為授權人向經濟部智慧財產局申請專利之附件或相關文件之一（專利申請案號：\_\_\_\_\_），請於 \_\_\_\_\_ 年 \_\_\_\_\_ 月 \_\_\_\_\_ 日後再將上列論文公開或上載網路。

因上列論文尚未正式對外發表，請於 104 年 8 月 22 日後再將上列論文公開或上載網路。

授權人：朱銘傑

親筆簽名及蓋章：朱銘傑

民國 100 年 8 月 26 日

電話：<sup>02) 2726-6863</sup> 0928-517-226 傳真：  
聯絡地址：台北市信義區大道里大道路13鄰58巷2號1樓  
E-Mail：henry760125@gmail.com

# 國立勤益科技大學

## 博碩士論文全文上網授權書

(提供授權人裝訂於紙本論文書名頁之次頁用)

本授權書所授權之論文為授權人在國立勤益科技大學  
電子工程系資訊科學組 99 學年度第 2 學期取得碩士學位之  
論文。

論文題目：嵌入式自動語音辨識系統應用於智慧型機器人  
指導教授：黃國興, 楊勝智

### ■ 同意

本人具有著作權之論文全文資料，非專屬、無償授予本人畢業學校圖書館，不限地域、時間與次數，以微縮、光碟或數位化等各種方式重製與利用，提供讀者基於著作權法合理使用範圍內之線上檢索、閱覽、下載及列印。

論文全文上載網路公開之範圍及時間：

校內區域網路	■ 立即公開
校外網際網路	■ 中華民國 104 年 8 月 22 日公開

授權人：朱銘傑

簽名：朱銘傑

中 華 民 國 100 年 8 月 26 日

國立勤益科技大學  
研究所碩士班  
論文口試委員會審定書

本校 電子工程系研究所 碩士班 朱銘傑 君  
所提論文 嵌入式自動語音辨識系統應用於智慧型機器人  
合於碩士資格水準，業經本委員會評審認可。

論文口試委員會：

召集人：許志剛

委員：黃國興

楊勝智

許志剛

指導教授：黃國興

所

長：

電子工程系主任 林熊鐵

中華民國 一 百 年 七 月

# 嵌入式自動語音辨識系統應用於智慧型機器人

指導教授：黃國興 博士

研究生：朱銘傑

共同指導教授：楊勝智 博士

國立勤益科技大學電子工程系研究所

## 摘要

本篇論文主要將自動語音辨識(ASR)系統應用於點餐服務機器人，利用 SPCE3200 開發板撰寫語音辨識的模組，使用此模組可以幫助統整語音資料。將語音辨識系統寫入嵌入式系統可提高機器人主系統的執行效率，減少資源佔用，且還能夠快速移植到其他機器人上進行作業。本篇論文中，先將聲音從麥克風輸入，經過 SPCE3200 內部的 A/D 轉換過後再進行取樣點的分析和音框的分割及加窗，再利用梅爾倒頻譜係數(MFCC)，通過 13 個不同頻率的三角濾波器組，優化訊號並減少背景雜訊，求得特徵參數。並以隱藏式馬可夫模型(HMM)作為辨識技術，利用維特比(Viterbi)演算法計算狀態內各個不同的機率值並選擇最佳路徑，最後計算總機率值和對結果。本篇論文的目的是要完成語音辨

識之隱藏式馬可夫模型設計，並且縮小實體面積，低能量，短時間之消耗之外，且需快速取得菜單與即時辨識之效果。本論文訓練了 20 組中文語音，利用維特比訓練演算法計算出期望值並且使用 k-mean 類聚演算法來整理這些經過訓練的語音資料庫，再將已收斂的模組儲存下來，當作樣本。聲音的輸出採用預錄的方式，將已完成的聲音儲存至 Flash 中，再進行組合並經由 SPCE3200 中的 D/A 轉換後進行播放。本篇論文測試三個不同的環境下各個餐點的字句辨識率，最後利用 MATLAB 模擬 MFCC 於每個取樣點的參數值。單詞最高的辨識率可達 92%，整體平均最高辨識率可達 82.6%，其整合測試的結果可靠度相當高。

# **The ASR Technique Based on Embedded System for Intelligent Robot**

Graduate Student: Ming-Jie Ju

Advisor : Dr. Guo-Shing Huang

Co-Advisor : Dr. Sheng-Chih Yang

Institute of Electronic Engineering  
National Chin-Yi University of Technology

## **Abstract**

This thesis is to apply ASR (Automatic Speech Recognition) technique to the meal service robot in order to the user can be easier to order the meal and increase interaction between the robot and human. So far, there is not any speech recognition module for the robot in the market. In general, modify slightly by using the voice technology and modules directly to be used for the robot. Therefore, this paper is based on Mel-frequency Cepstral coefficients (MFCC) to scratch the characteristic parameters and make use of Hidden Markov Model (HMM) as the recognition technology and create speech model by SPCE3200 for meal service robot. The main purpose of this thesis is to achieve the design of embedded HMM of speech recognition and reducing the small size, low power, short time consumption. In addition, it can fast finish training speech data and real-time recognition. This thesis presents to train 20 sets of Chinese speech data for the meal service robot. The voice output is using record. A sentence and multi sentences under the three different environments are need test. The highest recognition rate of a sentence has a successful rate 92% and the mean can have 82.6% with record in the smaller space. In summary, the integrated test results have a relevant reliability.



## 誌 謝

感謝指導教授黃國興、楊勝智博士，這兩年多來的細心栽培及指導，使得本論文得以順利完成，並感謝口試委員許佳興博士給予本論文之寶貴意見與悉心指正，使本論文益臻完善。

再則感謝林耕志學長在碩一時給予的指導與幫助，使我受益匪淺。感謝鍾毅宏學長、沈育龍學長，以及同學蕭順徽、高劭銓、陳柏承，學弟周冠榮、賴昱辰、羅仁彰在朝夕相處、相互勉勵下，使學習和思考方不斷的獲得成長，研究生活能多采多姿。

最後，謹將此碩士學位及論文獻給我最摯愛的雙親，感謝你們給我不虞匱乏的生活。再次感謝所有關心我的人，願你們與我分享這份喜悅和榮耀。

銘傑 謹誌於

國立勤益科技大學電子工程系研究所

# 目 錄

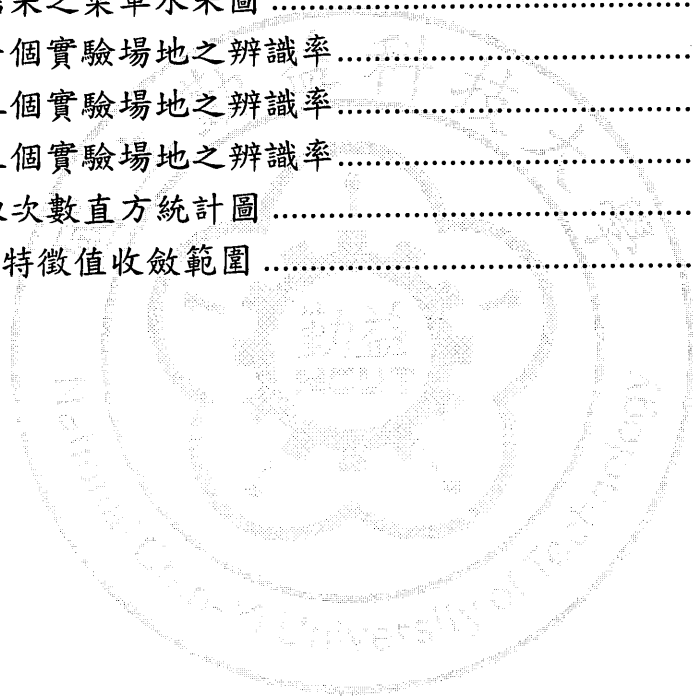
中文摘要 .....	I
英文摘要 .....	III
誌 謝 .....	IV
目 錄 .....	V
圖目錄 .....	VII
表目錄 .....	IX
第一章 緒論 .....	1
1.1 研究動機 .....	1
1.2 文獻探討 .....	3
1.3 本論文之貢獻 .....	5
1.4 論文架構 .....	6
第二章 系統與硬體架構 .....	7
2.1 智慧型機器人平台架構 .....	7
2.2 SPCE3200 開發版平台架構 .....	10
第三章 語音及特徵值求取方式 .....	14
3.1 訊號的取樣率及取樣解析度 .....	16
3.2 預強調 .....	19
3.3 能量偵測 .....	20
3.4 音框的分割 .....	23
3.5 漢明窗 .....	25
3.6 快速傅立葉轉換 .....	26
3.7 梅爾倒頻譜與三角濾波器 .....	28
3.8 能量對數 .....	31
3.9 離散餘弦轉換 .....	31
第四章 隱藏式馬可夫模型 .....	34
4.1 隱藏式馬可夫模型的基本概念 .....	36
4.2 隱藏式馬可夫模型產生結果機率的計算 .....	41
4.3 向前-向後演算法 .....	43
4.3.1 向前演算法之初始化 .....	44
4.3.2 向前演算法之遞迴 .....	44

4.3.3 向前演算法之結束 .....	44
4.4 向後演算法 .....	45
4.4.1 向後演算法之初始化 .....	46
4.4.2 向後演算法之遞迴 .....	46
4.5 隱藏式馬可夫模型的參數估算 .....	47
4.6 語音辨識的訓練演及分類算法 .....	50
4.6.1 訓練資料 .....	51
4.6.2 初始化中心值 .....	51
4.6.3 對元素進行分類 .....	51
4.6.4 更新類別的中心值 .....	51
4.6.5 判斷收斂與否 .....	52
4.7 隱藏式馬可夫模型最佳狀態序列的計算 .....	54
4.7.1 維特比演算法初始化 .....	56
4.7.2 維特比演算法之遞迴 .....	57
4.7.3 維特比演算法之結束 .....	57
4.7.4 維特比演算法之追朔路徑 .....	57
第五章 聲音的預錄及播放 .....	58
第六章 實驗結果 .....	60
第七章 結論與未來展望 .....	70
7.1 結論 .....	70
7.2 未來展望 .....	71
參考文獻 .....	72
作者簡介 .....	75

## 圖目錄

圖 1.1 機器人互動示意圖 .....	2
圖 1.2 語音辨識標準流程圖 .....	3
圖 2.1 智慧型服務機器人點餐介面.....	8
圖 2.2 智慧型服務機器人點餐介面 2.....	8
圖 2.3 機器人整體架構圖 .....	9
圖 2.4 智慧型機器人之系統架構圖.....	9
圖 2.5 SPCE3200 語音辨識系統開發版 .....	10
圖 2.6 EZJtag 燒錄器.....	11
圖 2.7 SPCE3200 結構圖.....	12
圖 2.8 SPCE3200 開發環境.....	13
圖 3.1 語音特徵值求取流程圖 .....	15
圖 3.2 聲音轉換系統圖 .....	16
圖 3.3 取樣解析度與輸出波形圖 .....	18
圖 3.4 預強調前信號.....	20
圖 3.5 預強調後信號.....	20
圖 3.6 經能量偵測後之信號 .....	21
圖 3.7 聲音訊號分割圖 .....	23
圖 3.8 漢明窗.....	25
圖 3.9 毫秒音波模擬圖 .....	27
圖 3.10 經快速傅立葉轉換之頻譜圖.....	27
圖 3.11 三角濾波器組 .....	29
圖 3.12 測試音譜圖.....	32
圖 3.13 通過三角濾波器之 MFCC 參數.....	32
圖 4.1 語音辨識概念圖 .....	35
圖 4.2 語音辨識流程圖 .....	35
圖 4.3 狀態數為四的由左向右馬可夫鏈.....	37
圖 4.4 狀態與狀態間之關係圖 .....	40
圖 4.5 向前演算法示意圖 .....	43
圖 4.6 向後演算法示意圖 .....	45
圖 4.7 狀態機率轉移演算示意圖 .....	48

圖 4.8 k-mean 於語音訓練結果之類聚圖 .....	52
圖 4.9 訓練演算法之流程圖 .....	53
圖 4.10 狀態與假設音框關係圖 .....	55
圖 5.1 聲音的預錄及輸出方塊圖 .....	58
圖 6.1 實驗結果之菜單主餐圖 .....	60
圖 6.2 實驗結果之菜單湯品圖 .....	61
圖 6.3 實驗結果之菜單飲料圖 .....	62
圖 6.4 實驗結果之菜單水果圖 .....	63
圖 6.5 實驗結果之菜單水果圖 .....	64
圖 6.6 於第一個實驗場地之辨識率 .....	66
圖 6.7 於第二個實驗場地之辨識率 .....	66
圖 6.8 於第三個實驗場地之辨識率 .....	67
圖 6.9 總訓練次數直方統計圖 .....	67
圖 6.10 語音特徵值收斂範圍 .....	68



## 表目錄

表 3.1 16 位元數據轉換表 .....	17
表 3.2 32 位元數據轉換表 .....	17
表 4.1 隱藏式馬可夫模型 $\lambda$ 狀態轉移機率矩陣 $A(7 \times 7)$ .....	38
表 4.2 隱藏式馬可夫模型 $\lambda$ 觀察結果轉移機率矩陣 $B(7 \times 64)$ .....	39



# 第一章 緒論

## 1.1 研究動機

國內唯一專業的機器人展覽「台北國際機器人展 TIROS(Taipei International Robot Show)」，近幾年來一直有企業不斷推出新的機器人，如：採智科技股份有限公司的「巡邏導覽機器人」、華寶通訊股份有限公司「互動陪伴型娛樂機器人」等。而這些機器人最大的共通點就是具有「人機互動」的優點，而可以辨識人類聲音的功能更是不能缺少。為了能夠研製出擁有可以辨識人類聲音的智慧型機器人，我們必須慢慢的訓練它，讓該機器人學會聽懂簡單的指令，並且利用語音回應，顯示在「使用者介面 UI(User Interface)」上，方便操作。語音辨識已經是逐漸實用化的技術，目前常用於 PDA、汽車導航系統或手機，但卻很少人將此功能應用於機器人上，所以本論文將語音辨識模組化，而功能集中設計在編號 SPCE3200 嵌入式系統內，不但美觀，同時也可縮小實際體積，方便操作。為了幫助使用者更容易與機器人產生互動，我們將此模組應用於智慧型點餐服務機器人身上，如圖 1.1 所示，不但能夠為使用者免除了拿筆手寫菜單的麻煩，還能在觸控界面上看到新鮮的餐點。

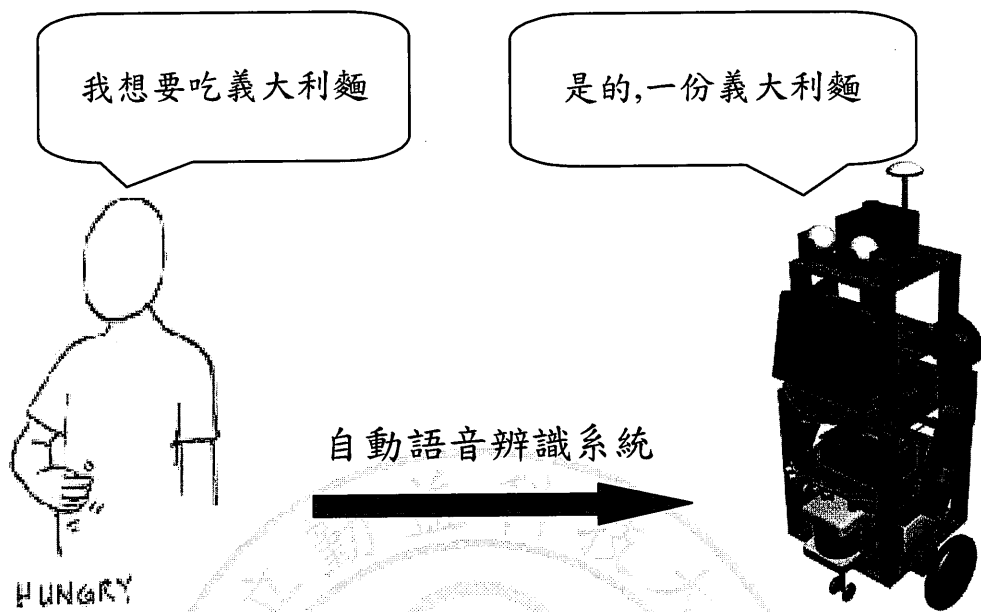


圖 1.1 機器人互動示意圖



## 1.2 文獻探討

人類藉由語言來做溝通，是最有效率的做法，因此我們在機器人之中加入了自動語音辨識系統[4]，讓機器人能夠簡單的分辨使用者需要的餐點，加入語音辨識的機器人能夠應用的範圍相當廣泛，例如居家照護，陪伴小孩，圖書導覽等。而嵌入式語音辨識系統提供一個快速、完整、方便的輸入，近年來也有頗多機器人採用類似的方法。而有不少人也看中這個原因，將嵌入式語音辨識系統用於智慧型手機[3]、移動式設備、家電、語音查詢、機器人[7]、車內導航[1]等。有些則辨識機械聲音的突波[2]，如煮沸的開水、門鈴、手機鈴聲等用來幫助聽覺障礙的人。甚至套入麥克風陣列[9]，用於音源辨識[15]等功能。這些都是應用於日常生活中的電子產品，而最關鍵的問題是要如何達到即時判斷、減少能量消耗與計算複雜度[1][6]、提高辨識度。

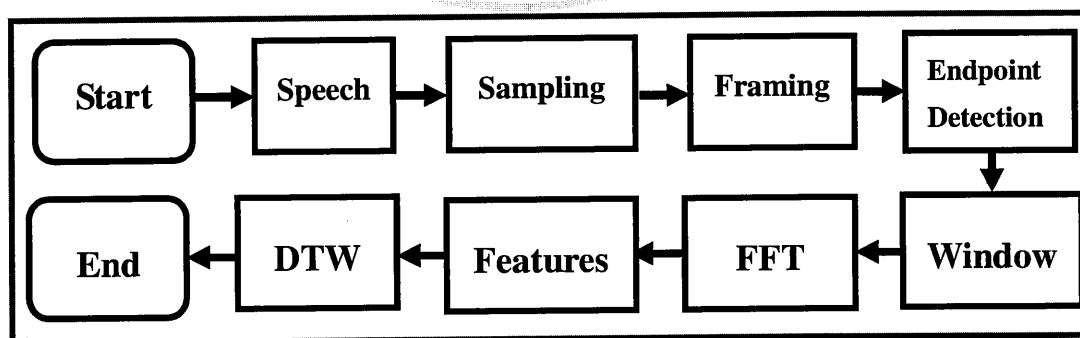


圖 1.2 語音辨識標準流程圖

圖 1.2 為普遍的語音辨識標準的流程圖，而內部需包括特徵值參數求取及主要辨別的演算方式。一般常用之語音特徵參數方法有 LPC(Linear Predictive Coefficients)[13]、MFCC(Mel Frequency Cepstrum Coefficients)[5][8]等。在辨識之前也同時需要大量參考樣本，以助提高辨識度，而語音訓練模型之方法有許多種，例如類神經網路(Neural Network)[14][18]、隱藏式馬可夫模型(Hidden Markov Model)[3]，而其中又以隱藏式馬可夫模型最佳[17]。辨識模組方法又可分為動態時軸校正(Dynamic Time Warping)[12]、隱藏式馬可夫模型(Hidden Markov Model)等，以動態時軸校正作為語音辨識只適合獨立詞辨識，因為他必須要做端點偵測，在起點與終點之間的語音訊號作比對。每一次語音輸入就得與所有語音參考樣本做比對，這會影響資源有限的嵌入式系統之大量計算複雜度與消耗時間，所以本篇論文採用隱藏式馬可夫模型作為辨識模組。利用隱藏式馬可夫模型為基礎的辨識方法，是目前為止最佳的語音辨識方法[17]，原因是他是以機率模型來描述發音的現象，將一小段語音的發音過程，看成是一個馬可夫模型中連續的狀態轉移。而狀態的轉移以及狀態下產生的結果，都是以機率統計的方式來做辨識的模型。

## 1.4 論文架構

本論文從最基本使用的系統與硬體架構開始說明，第二章節將會介紹智慧型機器人的設計以及整體的平台架構，並且展示出利用 Borland C++開發完成的點餐畫面供研究者參考，接下來對 SPCE3200 的嵌入式系統做簡單的內部介紹，以及使用的開發環境。第三章則是說明本論文所使用的語音及特徵值求取方式，這些使用方法在語音中被稱為前處理，分別為：預強調(高通濾波器)、能量偵測、音框切割、漢明窗、快速傅立葉轉換、梅爾倒頻譜與三角濾波器、能量對數、離散餘弦轉換，經過以上的過程就能取得音框中一連串的特徵參數，此方法被稱為梅爾倒頻譜。第四章是最後的演算法，名稱為隱藏式馬可夫模型，本論文將取得的特徵參數經過統計算法，並且假設多個參數於系統中，而這些假設的參數必須藉由訓練得知，最後以機率來表示一串字句的辨識率。第五章則介紹聲音的預錄及播放，我們取得許多不同菜單的聲音儲存在 SPCE3200 模組的 Flash 中進行合成聲音的播放。第六章為實驗結果，在三個不同的環境中做測試，並且對結果做統計，利用 MATLAB 畫出一小段時間的特徵參數。最後第七章結論與未來展望，此章節做最後的總結，針對未來研究與優點及改進的部分作探討。

## 第二章 系統與硬體架構

本論文的主旨在設計一嵌入式語音辨識系統於移動式機器人上，其功能包括語音辨識及合成輸出，使智慧型機器人能夠辨識使用者所選擇的餐點來做確認。將辨識後的結果傳至機器人主控端，並且在觸控螢幕作顯示。

本章將對智慧型機器人的架構及 SPCE3200 嵌入式開發版作詳細介紹，圖 2.1 為使用 Borland C++開發的主要點餐介面，圖 2.2 為點餐完成時總計的結果，圖 2.3 為本論文實驗室主要機器人的架構圖，圖 2.4 代表整體機器人的系統架構，語音系統也包含在其中。

### 2.1 智慧型機器人平台架構

論文以桌上型電腦(P5LD2)做為傳送及接收的控制中心，其應用的內容包括(1)觸控式螢幕之使用者介面(2)影像辨識系統(3)機械手臂系統(4)SPCE3200 語音辨識系統(5)避障系統(6)行走系統(7)導航系統。各個載具與主控端之間都經由 RS232 作資訊溝通的介面，並以 Borland C++作為主要整合的開發軟體。



圖 2.1 智慧型服務機器人點餐介面

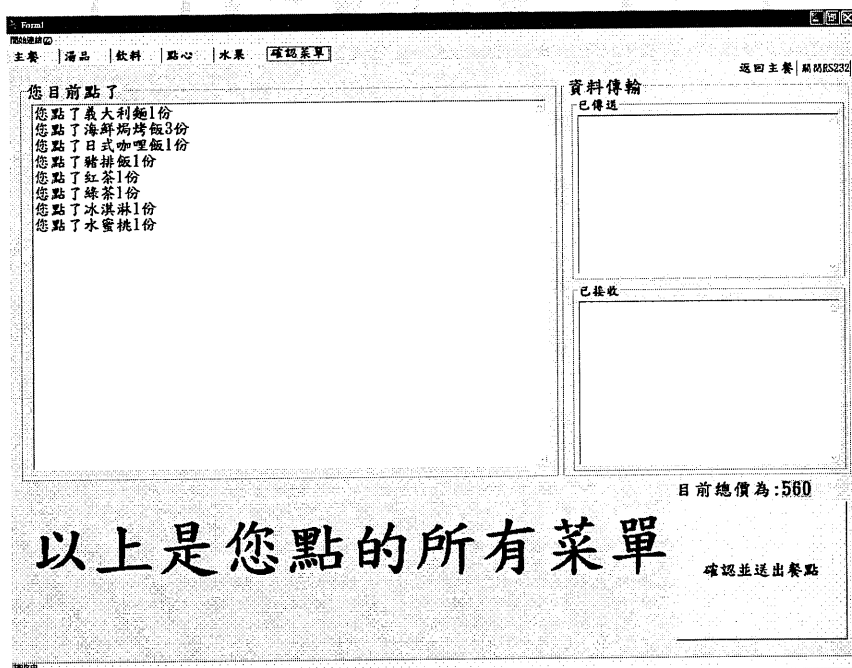


圖 2.2 智慧型服務機器人點餐介面 2

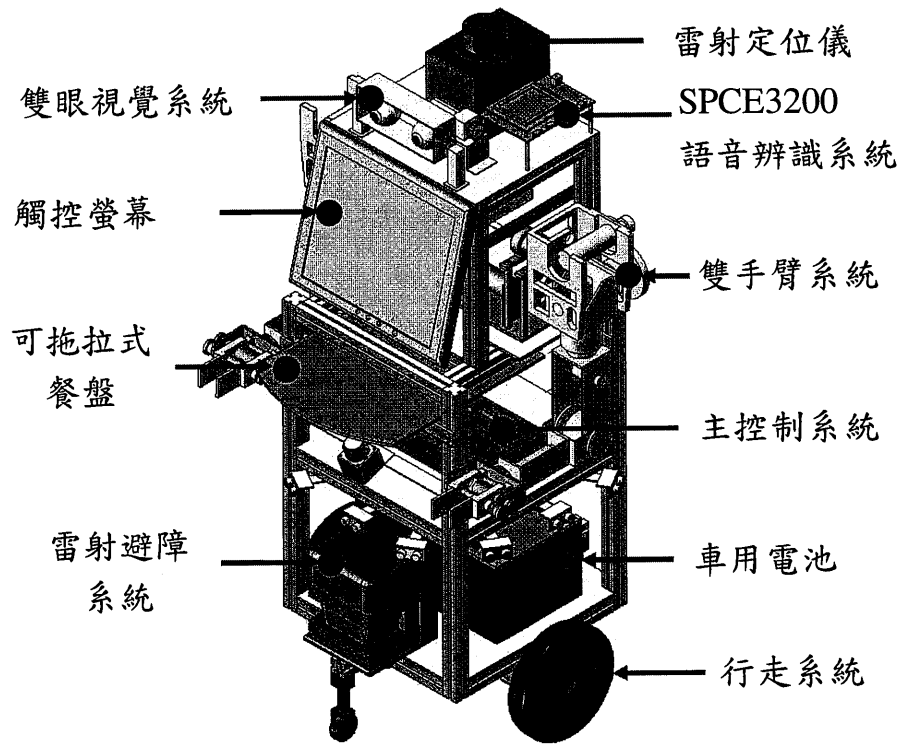


圖 2.3 機器人整體架構圖

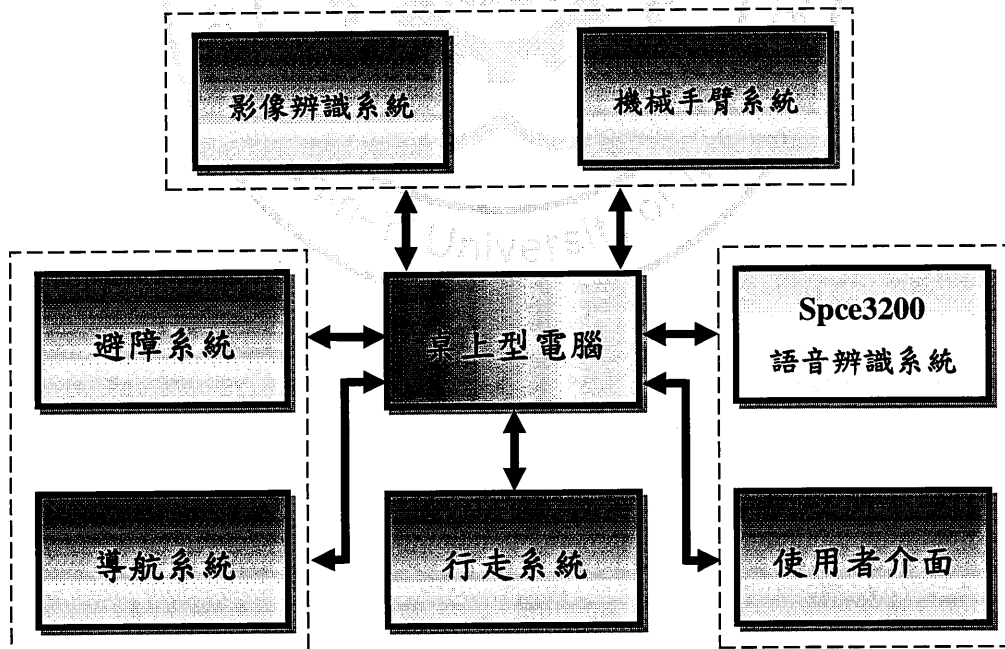


圖 2.4 智慧型機器人之系統架構圖

## 2.2 SPCE3200 開發版平台架構

在語音辨識部分，本論文採用 SPCE3200 開發版作為語音識別的撰寫平台，如圖 2.5 所示。此開發版為 32 位元，可嵌入作業系統，增加分配資源活用性。

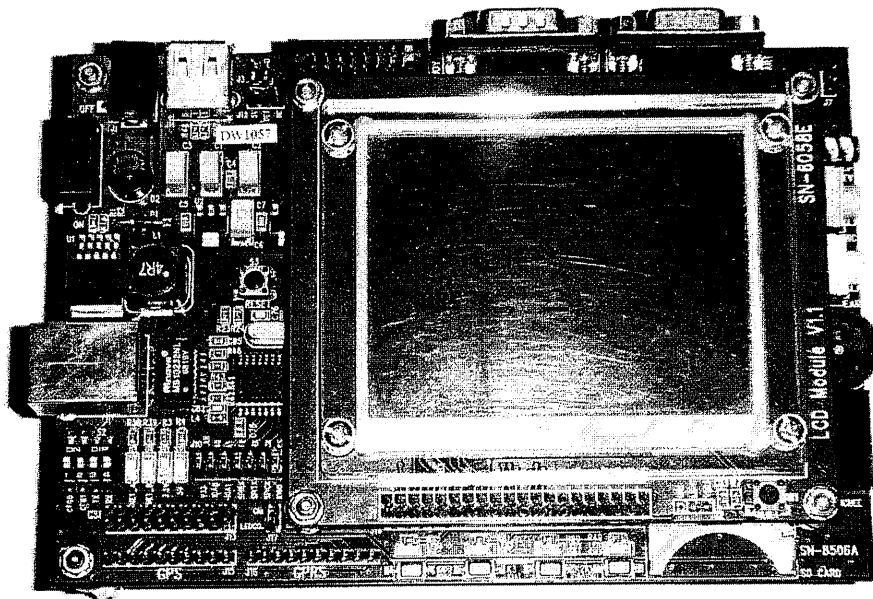


圖 2.5 SPCE3200 語音辨識系統開發版

SPCE3200 開發版具有的資源相當多，速度最快可達到 162MHz，如結構圖 2.7 所示。圖 2.6 為 SPCE3200 專用的 EZJtag 燒錄器，使用 print-port 做為接口。

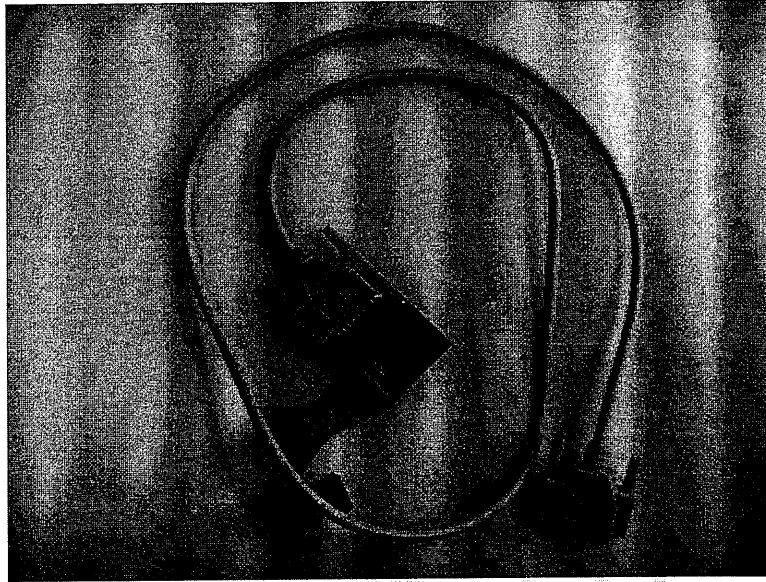


圖 2.6 EZJtag 燒錄器

記憶體: 128Mbits SDRAM、64Mbits NorFlash、128Mbits NandFlash

輸入模組: 麥克風輸入、4 個獨立按鍵、電阻式觸控螢幕

輸出模組: 4 個 LED、雙聲道音頻輸出、TFT LCD

通訊模組: UART、USB1.1、乙太網路

燒錄模組: EZJtag 燒錄器



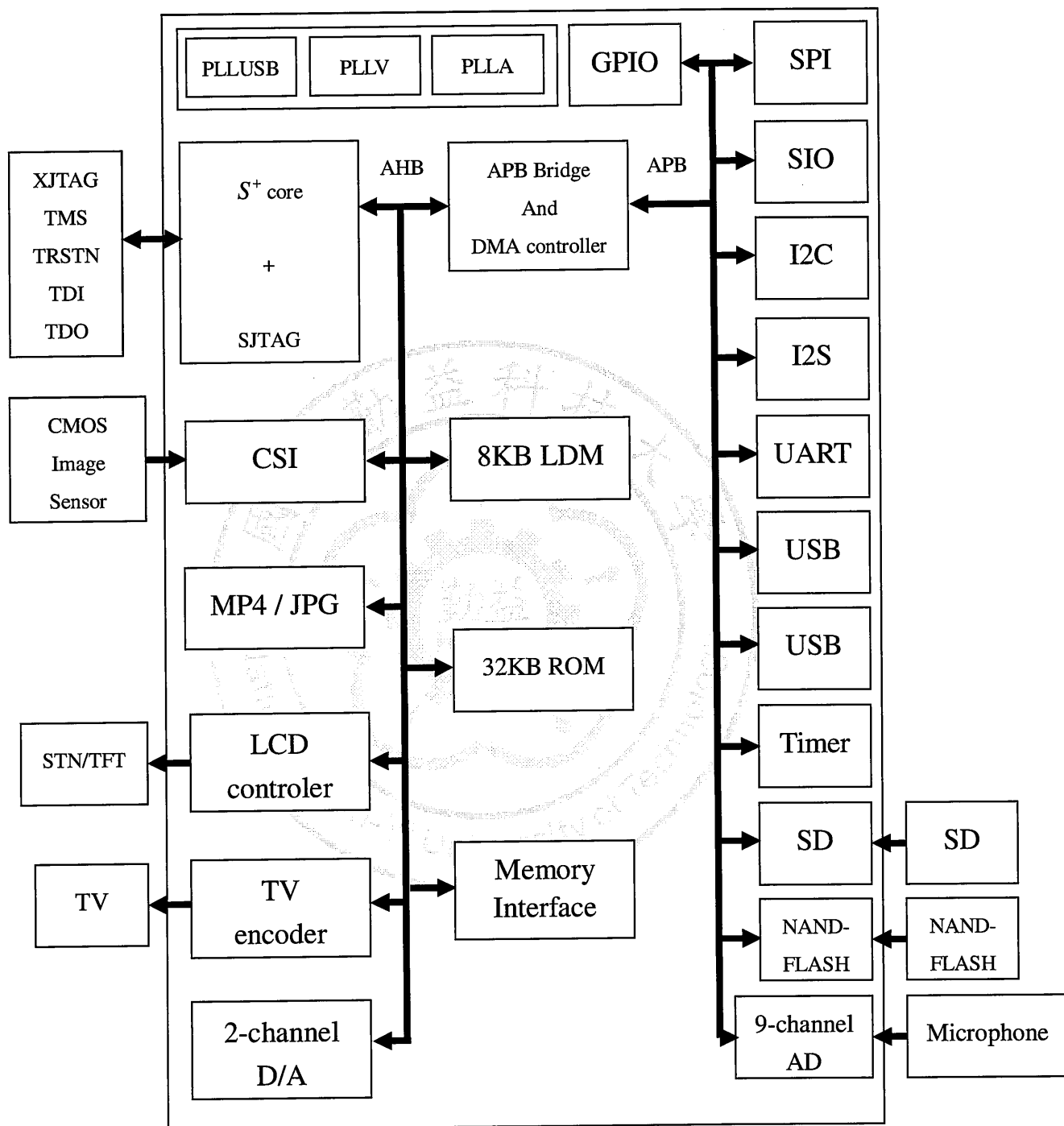


圖 2.7 SPCE3200 結構圖

SPCE3200 中有專用的開發環境，如圖 2.8 所示，提供使用者簡單容易操作的環境，提供單步執行及參數觀測等功能。

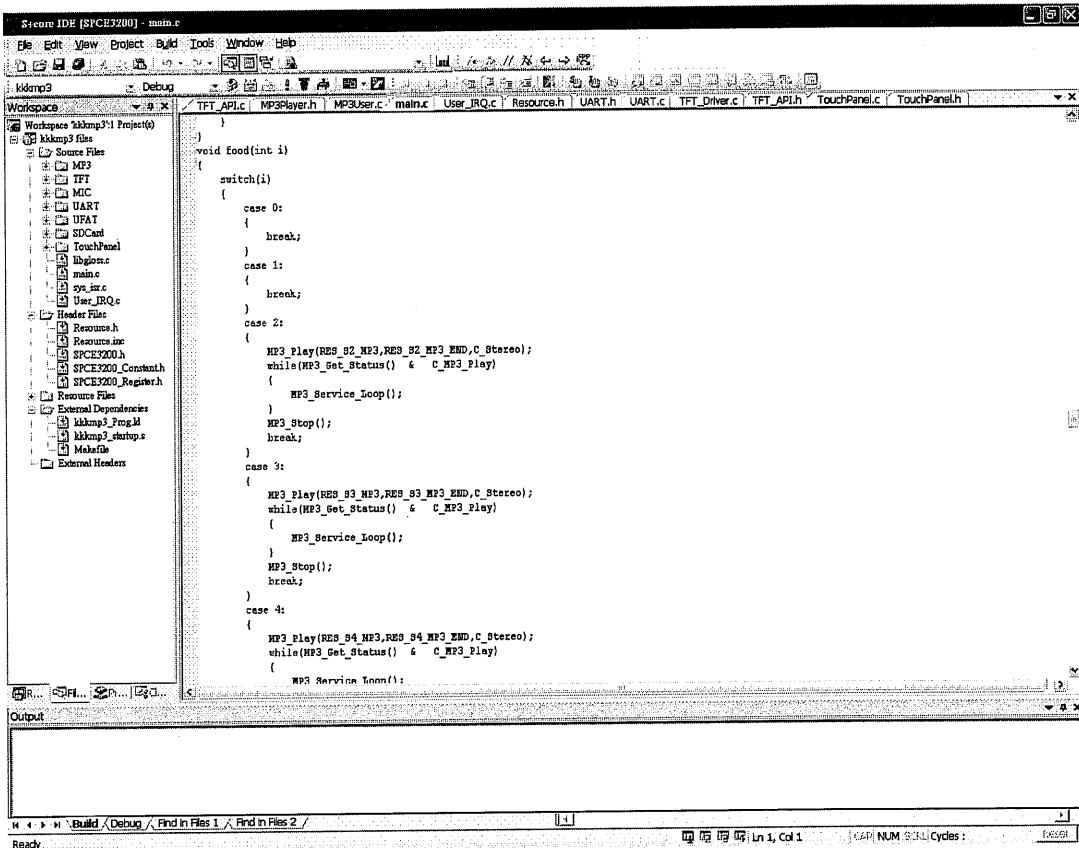


圖 2.8 SPCE3200 開發環境

在 SPCE3200 開發版中已有許多建立完成的函式庫能夠取得音訊進入麥克風時的信號，在電阻式觸控螢幕中也有相關的函式庫能夠即時套用，甚至也能夠支援一般 C 或 C++ 中的 math 函式庫供運算。此開發版也提供整合晶片的 BIOS，讓研究者不必煩惱硬體溝通的問題。

### 第三章 語音及特徵值求取方式

隨著人們對語音信號的研究，我們已經可以藉由語音和電腦做溝通了，也就是說機器人可以辨別出人類所說的話，因此想要操作機器人除了藉由鍵盤或滑鼠輸入之外，最直接的方法就是用口下命令。但在這之前我們必須從麥克風輸入的語音中求取特徵值參數。一般來說語音辨識可分為時域和頻域上的分析，兩者相比較之下，時域的計算量較小，但是辨識率相對來說較低，比如說像是線性預估分析(Linear Predict Coding)不需要經過頻域上的轉換。頻域上的分析就必須經過傅立葉轉換，此步驟的計算量較多，但是比較容易觀察出時間內頻率的變化量，相對的辨識率會較高。而本文採用的方法為梅爾倒頻譜係數，如圖 3.1 的流程圖所示，將一段時間內的語音信號進行處理並且取出有效的信號進行分析，以方便建立固定的單字模型。本章節將會對圖 3.1 流程圖中的每一個步驟詳細介紹，並且利用 MATLAB 進行模擬，將一些實際的信號進行處理後畫出，以便研究此論文者能夠更加了解。

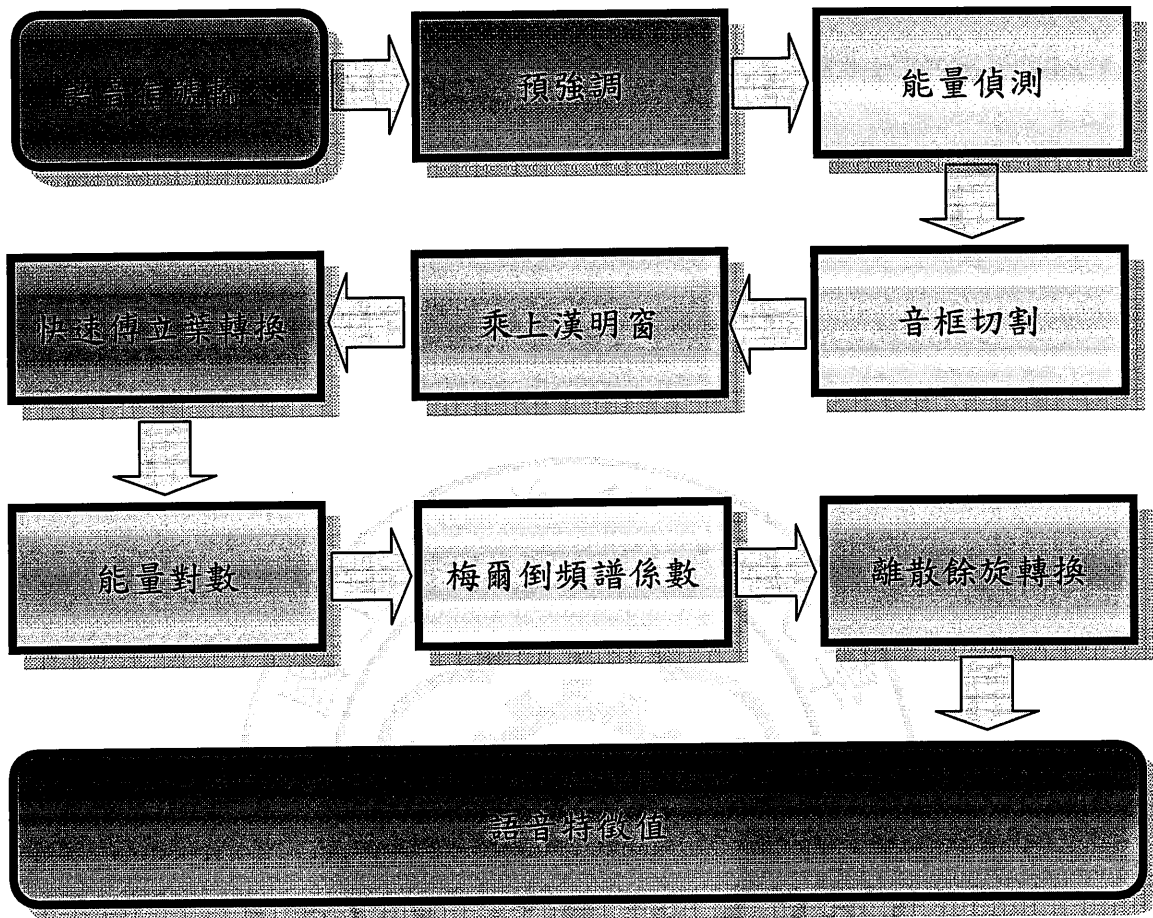


圖 3.1 語音特徵值求取流程圖

### 3.1 訊號的取樣率及取樣解析度

本論文聲音的輸入是經由麥克風，而在嵌入式系統中，聲音不再像是以往錄音裝置一樣儲存類比波形，而是改為一連串的數據儲存及處理操作，這串數據代表音調高低或說話的大小聲。通常麥克風接收到訊號之後會先經過初步的處理，如圖 3.2 所示，例如：濾波、ADC(Analog to Digital Converter)轉換，就可以將聲音的波形轉化為數據資料。經由麥克風輸入的每一筆資料都是經過一個取樣的動作所得到的。在一秒中所做的取樣次數就被稱為取樣率。

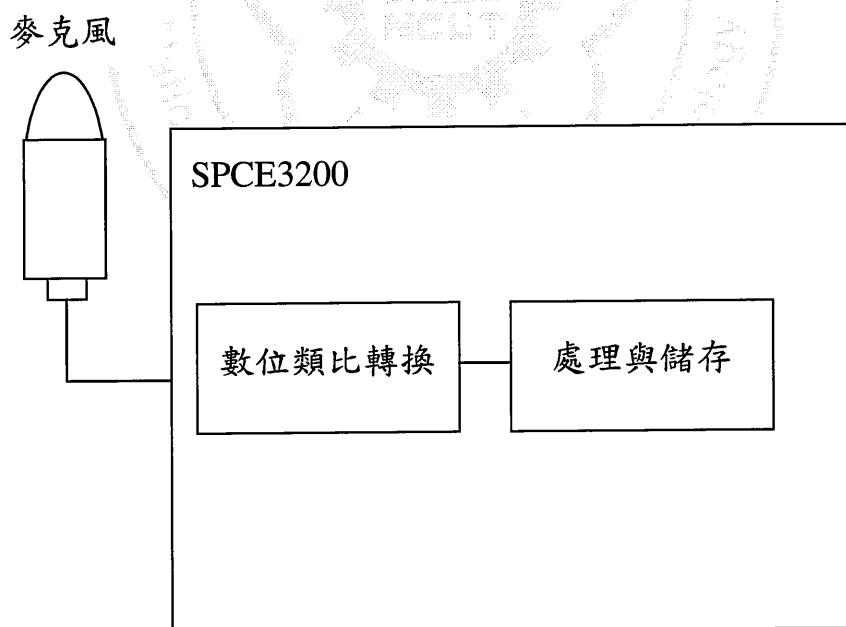


圖 3.2 聲音轉換系統圖

通常數據是以二進制的方式呈現的，舉一個簡單的例子，如表 3.1 所示。

以本論文的 SPCE3200 為例，由於 SPCE3200 為 32 位元，所以轉換出來的數據為表 3.2 所示。

表 3.1 16 位元數據轉換表

二進制	十進制	十六進制
1111111111111111	65535	FFFF

表 3.2 32 位元數據轉換表

二進制	十進制	十六進制
1111111111111111 1111111111111111	4294967295	FFFFFFFF

取樣的解析度決定了接收聲音的品質，也決定了數據資料的大小。若將類比波形看成是一個滑面斜坡，而這個斜坡被數位化之後就會成為沿著斜坡建立的階梯一般，這個斜坡的階梯數越多，就會與原本的波形越相近，如圖 3.3 所示，如果把解析度加大，以 SPCE3200 為例，用 12 位元的轉換器，那就會有 4096 個階梯，每增加一個位元，取樣

階梯數就會加倍。

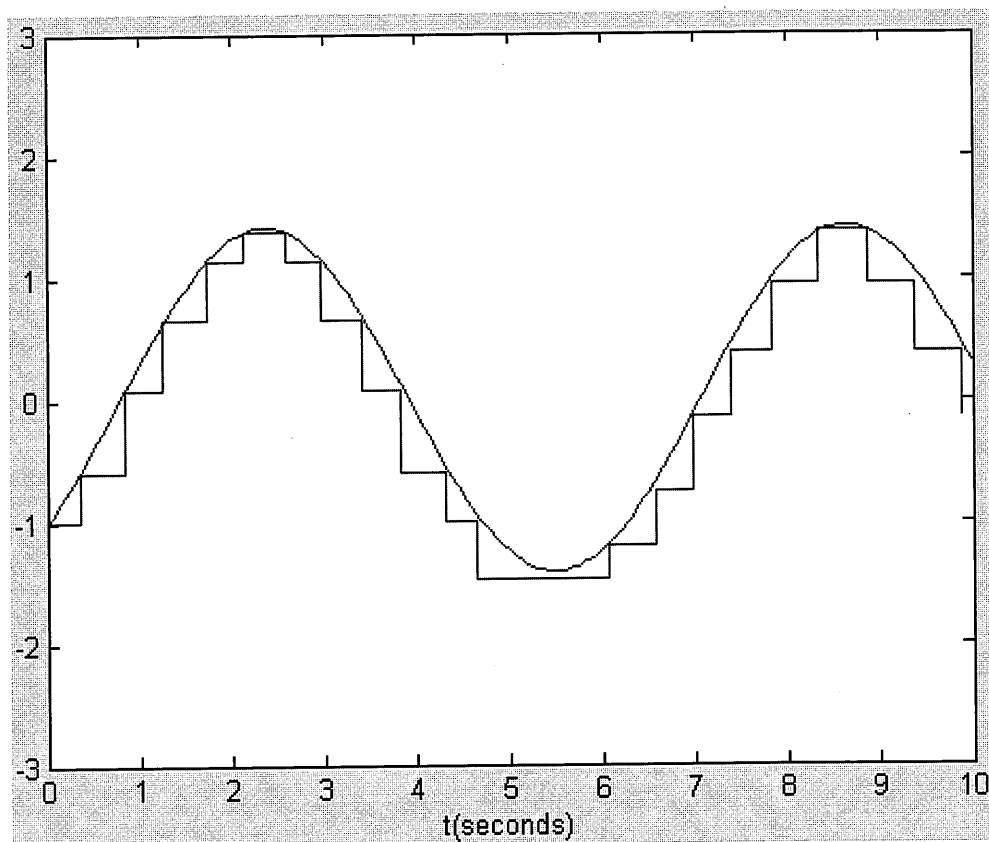


圖 3.3 取樣解析度與輸出波形圖

用什麼樣的取樣速度時有時會比取樣解析度更重要，假如取樣率遠低於捕捉訊號的最大頻率，那麼就有可能發生整個訊號完全失真的情況，這種情況就很像是訊號無緣無故憑空消失，這樣的數據就無法進行處理。假如取樣率剛好是與被取樣訊號的頻率相同，那麼將會得到一條直線的數據，那是因為取樣地點都發生在訊號波形上的同一個地方，如果取樣率是輸入訊號頻率的兩倍整，也會發生同樣的情況，為

了避免這種情況發生，因此取樣率必須要比輸入訊號頻率的兩倍還高一些。就聲音而言，人類的聽覺器官可以感知的頻率範圍為 16Hz 到 18kHz 的聲音，而說話頻率大概是 80Hz 到 1kHz 之間，以 SPCE3200 為例子，本篇論文所使用的取樣頻率為 16kHz，而取樣頻率使用 16KHz 的原因在於能夠更仔細的觀察時間內聲音波形的變化量，但相對的資料量也會增加。

### 3.2 預強調

聲音從人類嘴唇發出後，經由麥克風輸入至處理系統時，聲音高頻的部分會被衰減掉，所以本論文在框與窗的切割及處理前，需要做高頻的補償，這個方法就是讓信號通過一個高通濾波器。

$$S(n)' = S(n) - \alpha * S(n-1) \quad (3.1)$$

$S(n)$  為原始信號， $S(n)'$  為預強調後的信號， $\alpha$  的值可由 0.9~0.99 之間，最佳值於 0.94~0.95。我們在下面舉一個簡單圖片的例子：



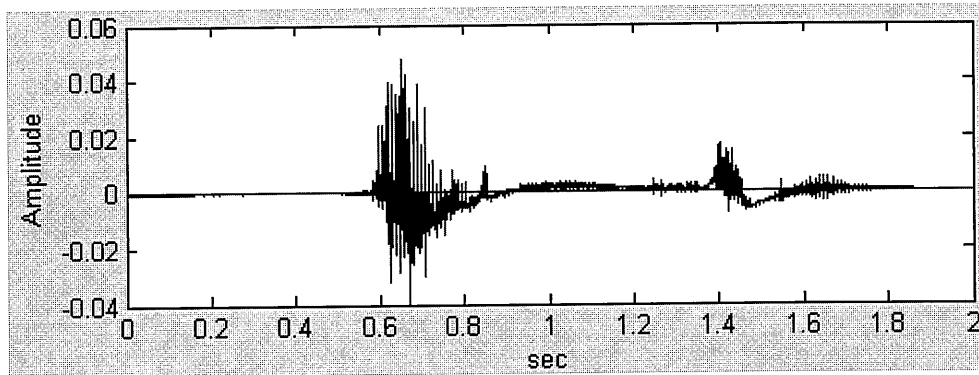


圖 3.4 預強調前信號

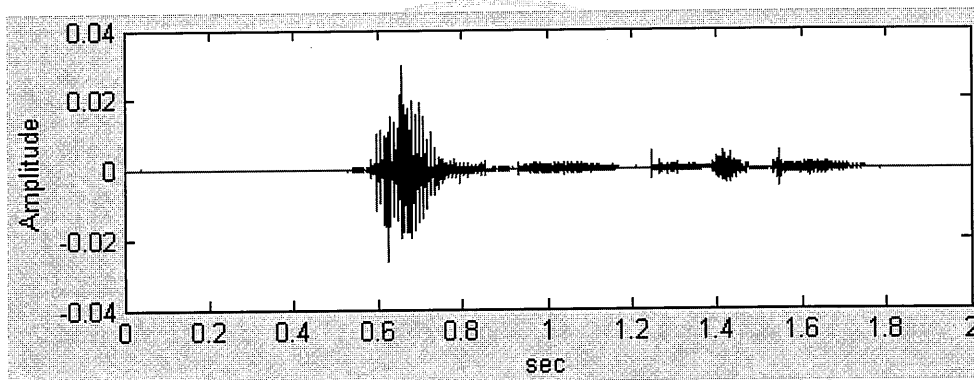


圖 3.5 預強調後信號

圖 3.4 為原始信號，也就是預強調前之信號，圖 3.5 為預強調之後的信號，可以很明顯的觀察出兩者之間的差別，信號明顯平穩許多，部分的背景雜音也被去除，高頻部分也被補償回來。

### 3.3 能量偵測

在做音框分割的處理之前，我們必須先區分有聲段與無聲段的部分，然後才會利用有聲的部分來進行識別。而大部分使用的方法為能

量偵測，是在時域中進行的演算法，計算部分較為容易，對於嵌入式系統也不會造成過多的負擔。首先我們必須設定一個 Threshold，讓雜訊或無聲段的部分消失，以便分割訊號。

$$\text{Threshold} = 8\% \times \max[E(n)] + \frac{1}{P} \sum_{i=1}^P E(i) \quad (3.2)$$

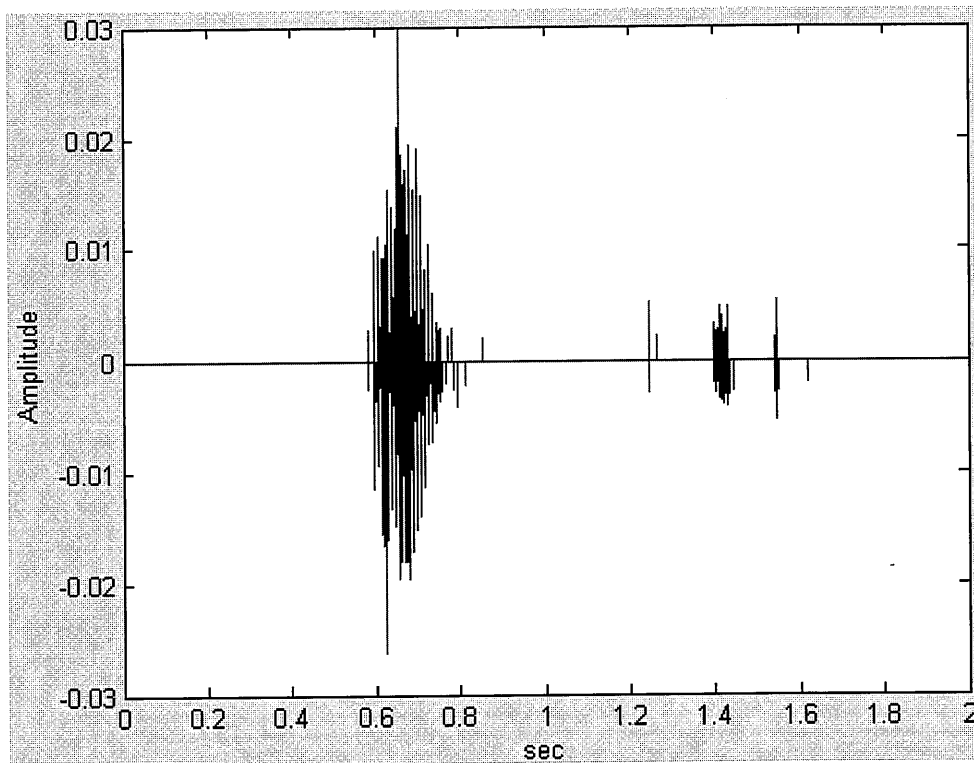


圖 3.6 經能量偵測後之信號

大部分能量偵測的門檻值都會將信號中的最大值乘上  $\frac{1}{32}$  當作門檻值，但是當信號中的雜訊不穩定時，建立此門檻值就沒有意義了。所以我們採用公式(3.2)，此方法能夠對應最合適的場地來建立門檻值。

$\max[E(n)]$  代表一段時間內有聲音框內的最大值，其值超過正常語音訊號的值，乘上 8% 後加上一連串雜訊  $E(i)$  的平均值，而 P 為一段雜訊音框的時間長度，乘上 8% 的意義在於需要以較大的信號來啟動切割的起始點，這些參數需要從嵌入式系統截取信號後進行設定，可由圖 3.6 觀察。若取得到的信號大於 Threshold，那麼就會被留下，並代表語音的起始點，而結束點也是藉由此 Threshold 進行判斷。



### 3.4 音框的分割

一般我們都會以正常的速度講話，很少有連續一口氣說了好幾個字，中間至少都會有少許的停頓，以便能容易地將意思表達清楚。在做語音辨識上，便是利用存在語音間的這些片斷靜音將真正的語音作切割來分析，否則說了一整串話，我們無法下手來處理和比對。語音信號切割的好壞，直接影響辨認的結果，因為一段切錯了的語音信號，將很難有辨識正確的機會。

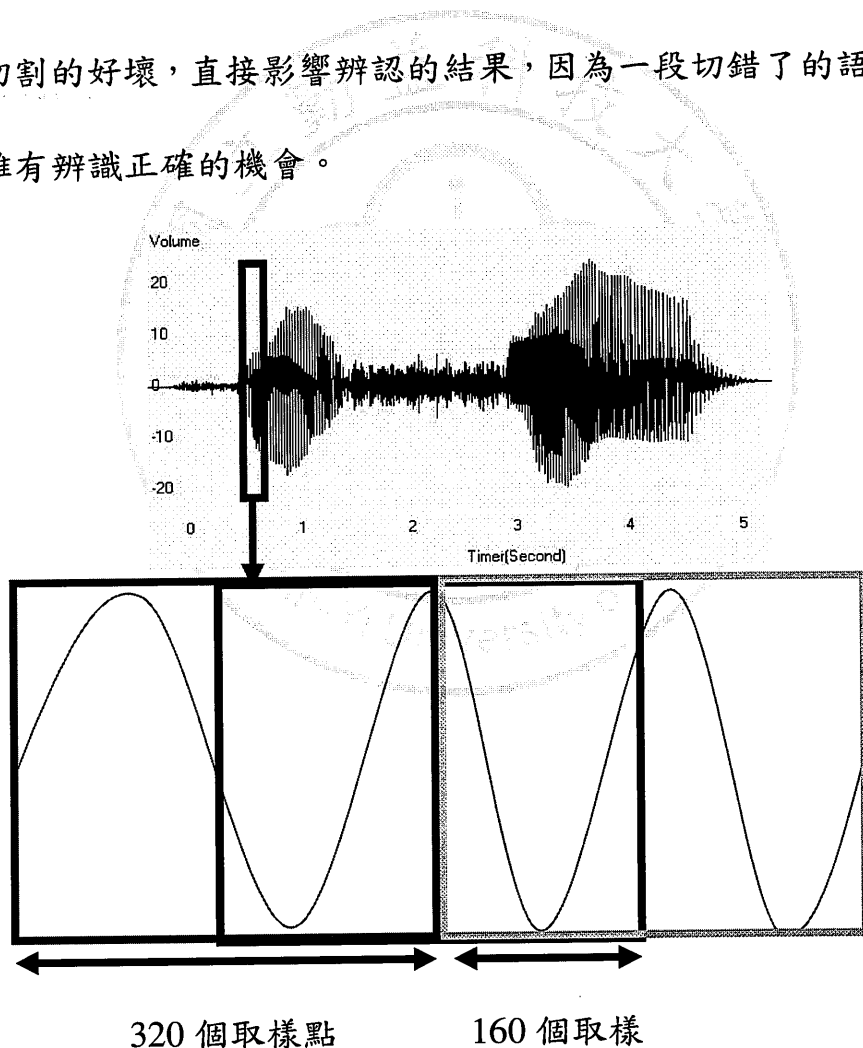


圖 3.7 聲音訊號分割圖

通常音框的長度以 10ms 到 40ms 為主，如圖 3.7 所示。以本篇論文的取樣率 16kHz 為例，解析度為 12bits(1.5byte)，在 10ms 內就有 160 個取樣點，100ms 就有 1600 個。而以聲音檔案大小來算，若截取了 2 秒的聲音，取樣點就為 32000，那麼大小就為  $32000 \times 1.5$  等於 48000 個點，大小就為 48KB 左右，由此可見語音檔案在嵌入式系統中的大小。在本篇論文分割的音框為 20ms，重疊 10ms，最後音框取樣點的計算方式如下：

音框切割範圍 = 取樣頻率 \* 20ms / 1000ms = 320 points

音框重疊部分 = 取樣頻率 \* 10ms / 1000ms = 160 points

音框跳距 = 音框切割範圍 - 音框重疊部分

音框率 = 取樣頻率 / 音框跳距(160 points) = 100 points/s

音框切割範圍代表 20ms 中所需要的音訊部分，音框重疊部分代表音框與音框切割的範圍中重疊的部分，音框跳距為此音框切割的起始點到下一個音框起始點的距離，音框率為每秒出現的音框數目。

我們認為語音單字的長度至少也有 200ms(若以很快的速度說話)，所以音框長度小於 10ms 的語音則放棄，因為這很有可能是由環境雜音所引起的，而不是真正的語音信號。

### 3.5 漢明窗

視窗有兩種選擇，一種是矩形窗(rectangular window)與漢明窗。當語音信號被乘上矩形窗時，窗內的語音訊號會被保留，但視窗外的就被當作為 0，這會造成視窗兩邊訊號的不連續，就好像是將語音訊號接一段下來一樣。而乘上漢明窗的主要目的是為了要加強框與框的連續性連續性，那是因為本論文假設進行快速傅立葉轉換時，音框內都是連續的週期訊號，如果這個週期性不存在，就會造成轉換上的誤差資料。漢明視窗的長度決定了特徵參數的準確性，長度越長，用來計算語音特徵參數的取樣點也相對的越多，但有時候會有被平均掉的現象。因此語音訊號變化快的時候，視窗取的短，便比較容易看出短時間內的變化，圖 3.8 為本論文所使用的漢明窗。

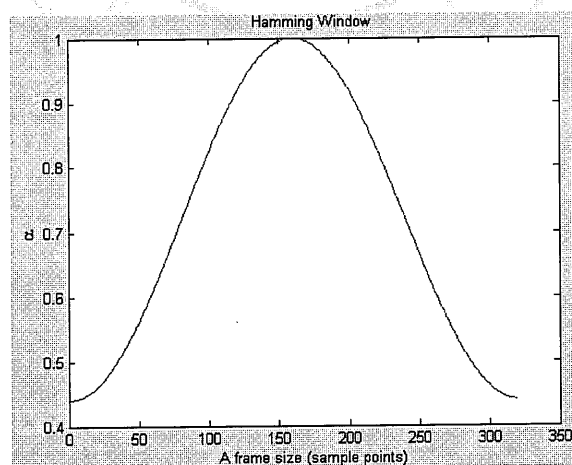


圖 3.8 漢明窗

圖 3.8 之  $\alpha$  值為 0.46。

$$w(n) = \begin{cases} 0.54 - 0.46 \cos\left(\frac{2n\pi}{N-1}\right), & 0 \leq n \leq N-1 \\ 0, & \text{otherwise} \end{cases} \quad (3.3)$$

$w(n)$  為一段時間長度的漢明窗。N 為視窗的長度，而  $n$  為時間長度內的取樣點數為 320 個點，其  $n$  參數不得大於漢明窗長度 N，如此才可完整的將語音訊號包覆於窗內。

### 3.6 快速傅立葉轉換

快速傅立葉轉換是離散傅立葉轉換的快速算法，一般的語音信號在時域時資料量非常龐大，所以我們將這些資料經由快速傅立葉轉換至頻域方便作分析及計算。

$$F(m) = \sum_{F=0}^{F-1} p(n) \cdot w(n) \cdot e^{-jn\omega_0 t}, \quad \omega_0 = 2\pi/T_s \quad (3.4)$$

$p(n)$  為傅立葉係數， $w(n)$  為漢明窗。在下面我們舉一個簡單的例子。

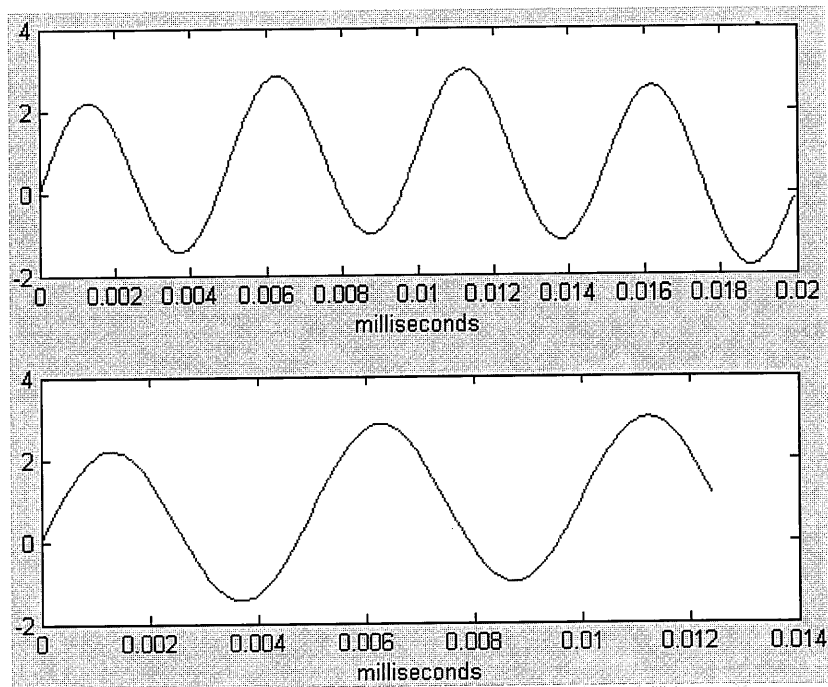


圖 3.9 毫秒音波模擬圖

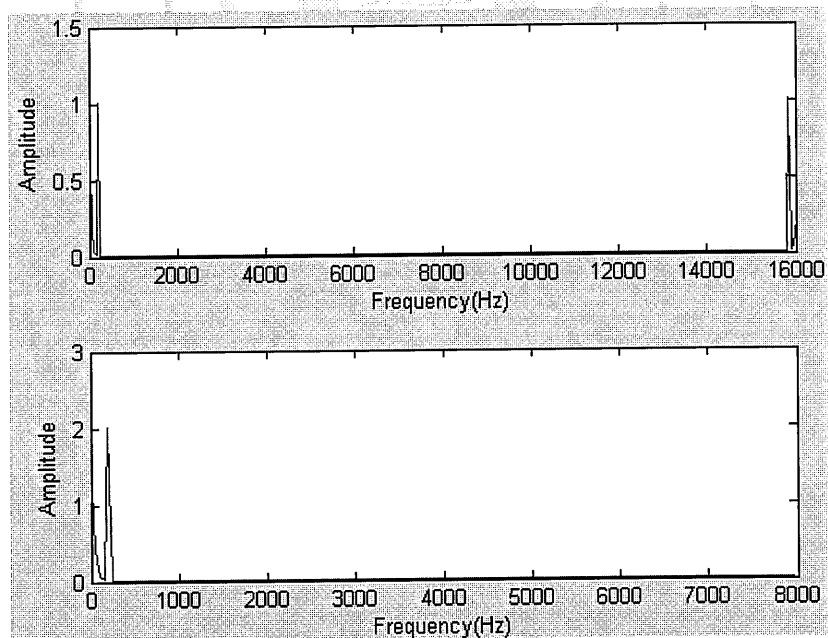


圖 3.10 經快速傅立葉轉換之頻譜圖



圖 3.9 為模擬一小段聲音波形的圖片，經過快速傅立葉轉換過後可得到圖 3.10 的頻譜圖，模擬中所使用的取樣頻率為 16kHz。

這些音波經過轉換後可以明顯的取得音框內頻域的成分大小，以方便計算分析。

### 3.7 梅爾倒頻譜與三角濾波器

在語音信號分析中，我們常常會將時域中的波形經過傅立葉轉換成頻域中，在頻域中觀察語音訊號的現象。如果將頻域中各個頻率成分取對數值，然後再做逆向的傅立葉轉換，回到時域中，就能得到一組新的參數，叫做倒頻譜係數。由於人類的聽覺系統對於一般語音訊號的相位並不靈敏，所以語音的頻譜中還是以能量為主。人類的耳朵並非對全頻域都有相同的敏感度，而正常來說在低頻的時候能夠比較容易分辨出頻率的差異。而通常在 1kHz 以下的部分會以 100Hz 做為頻帶寬度，1kHz 以上會以指數增加。因此本論文可以配合人類的聽覺特性，在頻域中將梅爾量度劃分頻帶，再將這些頻率成分合在一起看成一個能量，最後經過離散餘弦的演算轉換為倒頻譜。

首先本論文先介紹三角濾波器組[10]:

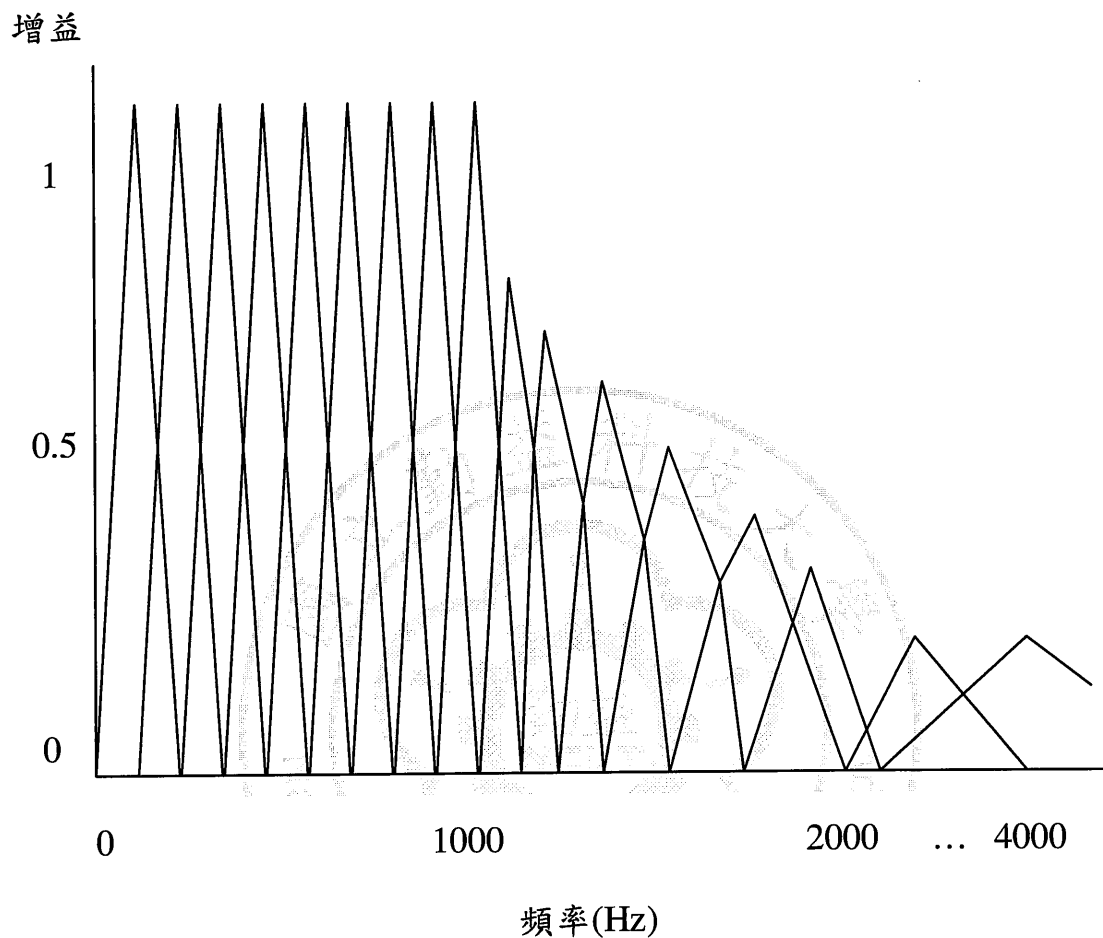


圖 3.11 三角濾波器組

如圖 3.11 所示，我們將 4kHz 以下的頻率分為 20 個頻帶，此濾波器組能夠隨著頻帶寬度而減少增益，能夠使高頻能量經過時不被放大，而三角形的面積不變。

$$M = 2595 \cdot \log_{10} \left( 1 + \frac{f}{700} \right) \quad (3.5)$$

這裡的 M 指的是在頻帶 f 之下有 M 個三角濾波器。

公式(3.5)為頻率以及梅爾之間轉換的方式。

$$B_m(k) = \begin{cases} 0, & k < f_{m-1} \\ \frac{k - f_{m-1}}{f_m - f_{m-1}}, & f_{m-1} \leq k \leq f_m \\ \frac{f_{m+1} - k}{f_{m+1} - f_m}, & f_m \leq k \leq f_{m+1} \end{cases} \quad (3.6)$$

$B_m(k)$  代表第 m 個頻帶的三角濾波器， $f_m$  為第 m 個頻帶的中心頻率，

$f_{m-1}$  與  $f_{m+1}$  為前後相鄰兩個頻帶的中心頻率，M 是全部頻帶的數目。

經過以上的演算法之後還須經過標準化的過程。

$$A_m = \frac{A}{f_{m+1} - f_{m-1}} \quad (3.7)$$

$A_m$  為演算後的增益，也就是圖 3.11 中的三角濾波器的高度，可將參數

A 設定的範圍為 180~240。

一般來說，正規化過後的三角濾波器有 20 組左右，但本論文只用到前 13 個三角濾波器，其中心頻率為 100、200、300、400、500、600、700、800、900、1000、1148、1318、1514。因為此頻段比較能夠接近人類對語音的區分性，而最高頻段大約落在 1500 左右。

### 3.8 能量對數

$$Y(m) = \log \left\{ \sum_{k=f_{m-1}}^{f_{m+1}} |F(m)|^2 B_m(k) \right\} \quad (3.8)$$

將這些經過 FFT 的個頻率能量  $|F(m)|^2$  乘上以上所敘述的三角濾波器，然後累加起來取對數值就為對數能量  $Y(m)$ 。

### 3.9 離散餘弦轉換

$$C[n] = \frac{1}{M} \sum_{m=1}^M (Y(m)) \cdot \cos \left[ \frac{\pi n(m - \frac{1}{2})}{M} \right] \quad (3.9)$$

$M$  為三角濾波器的個數， $C[n]$  為對數能量經過離散餘弦轉換後的語音音框中 MFCC 第  $n$  個參數。我們在下面舉一個例子，如圖 3.12、圖 3.13 所示。

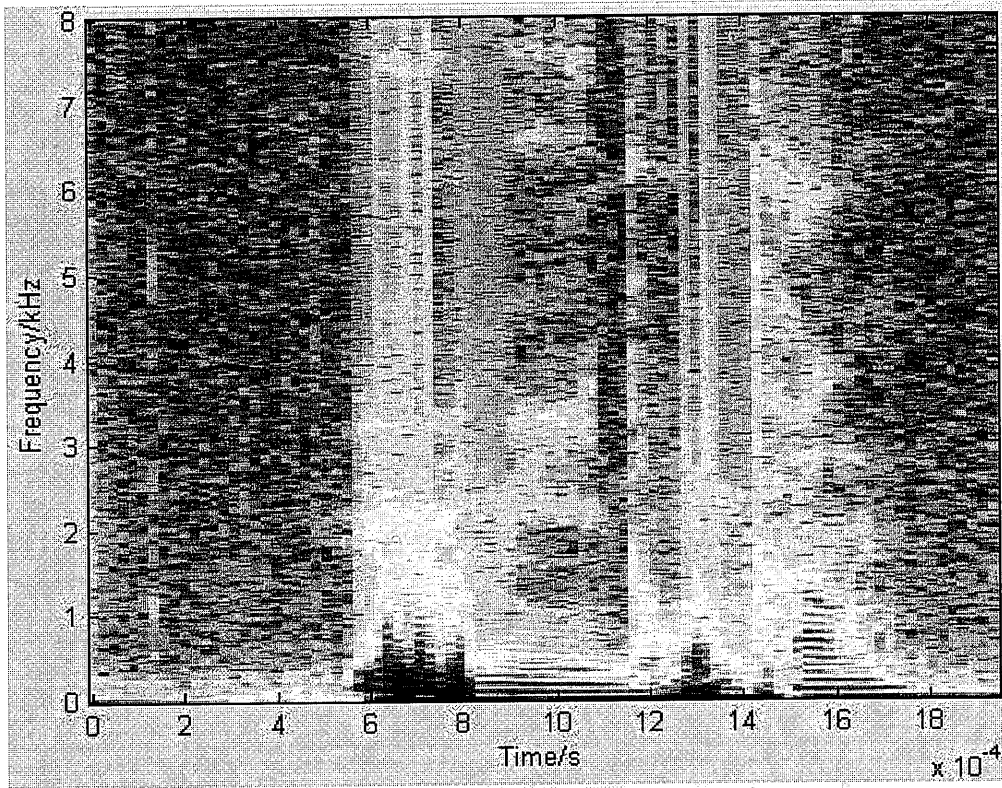


圖 3.12 測試音譜圖

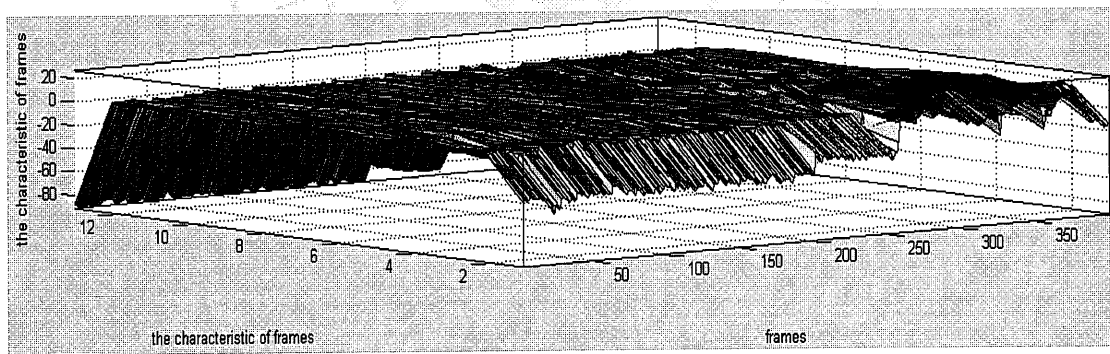


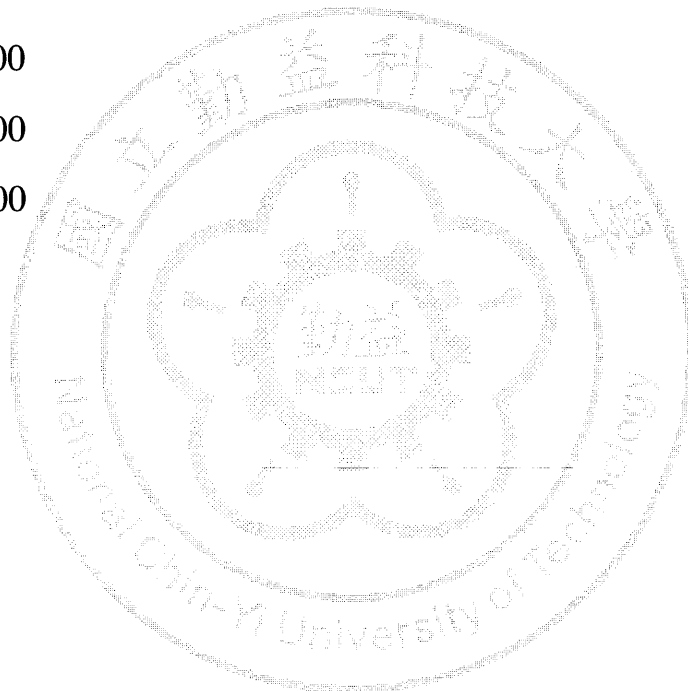
圖 3.13 通過三角濾波器之 MFCC 參數

圖 3.12 為本論文測試的音譜圖，取樣頻率為 16kHz，總時間約為 2 秒，為男性的一段聲音。圖 3.13 為通過三角濾波器之 MFCC 參數。經過上述的各種演算法將音框內的特徵參數取出後如圖 3.13，由於使用前 13 個三角濾波器，所以每一個音框所取得的特徵參數為 13 個，絕大多數多為浮點數。於 MATLAB 測試下的部分參數值：

6.157622e+000

8.063761e+000

2.450354e+000



## 第四章 隱藏式馬可夫模型

HMM 在做語音辨識時，需要先建立語音模型。他以機率模型來描述發音，將一小段語音的發音過程看成是一個馬可夫模型中連續的轉移狀態。再狀態的轉移下產生的結果，都以機率模型來描述。而在一個音框中呈現的語音特徵參數，看成是某一個狀態下的結果。也就是說將一連串的隱藏式馬可夫模型串接起來就代表一段語音。本論文採用維特比演算法於隱藏式馬可夫模型中進行辨認及訓練並使用 k-mean 類聚演算法整理可用的語音資料。首先我們必須將取得的 13 個特徵向量量化，採用的是行向量，並且給與這些量化過的向量一個字碼，字碼的部分可自行建立。我們將欲訓練的語音模型樣本經過估算的機率值，對每一個中文單字建立一個隱藏式馬可夫模型  $\lambda$ ，此參數為每一個中文單字的參考模型，若有 N 個單句即建立 N 個隱藏式馬可夫模型。在辨識的部分，由麥克風輸入的語音信號經過處理之後，在處理音框的特徵值向量最後轉換成相對的字碼，在經由向前程序之演算法求出機率，並且比對完所有單句模型之後，選擇出最大機率的模型及為最後的辨識結果，辨識概念與流程圖如圖 4.1 及圖 4.2 所示。

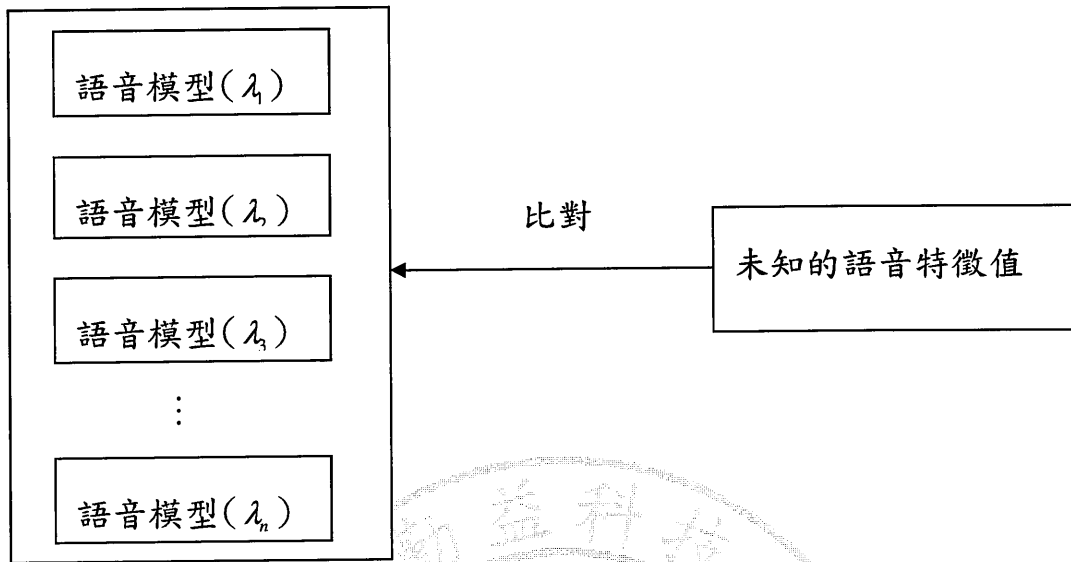


圖 4.1 語音辨識概念圖

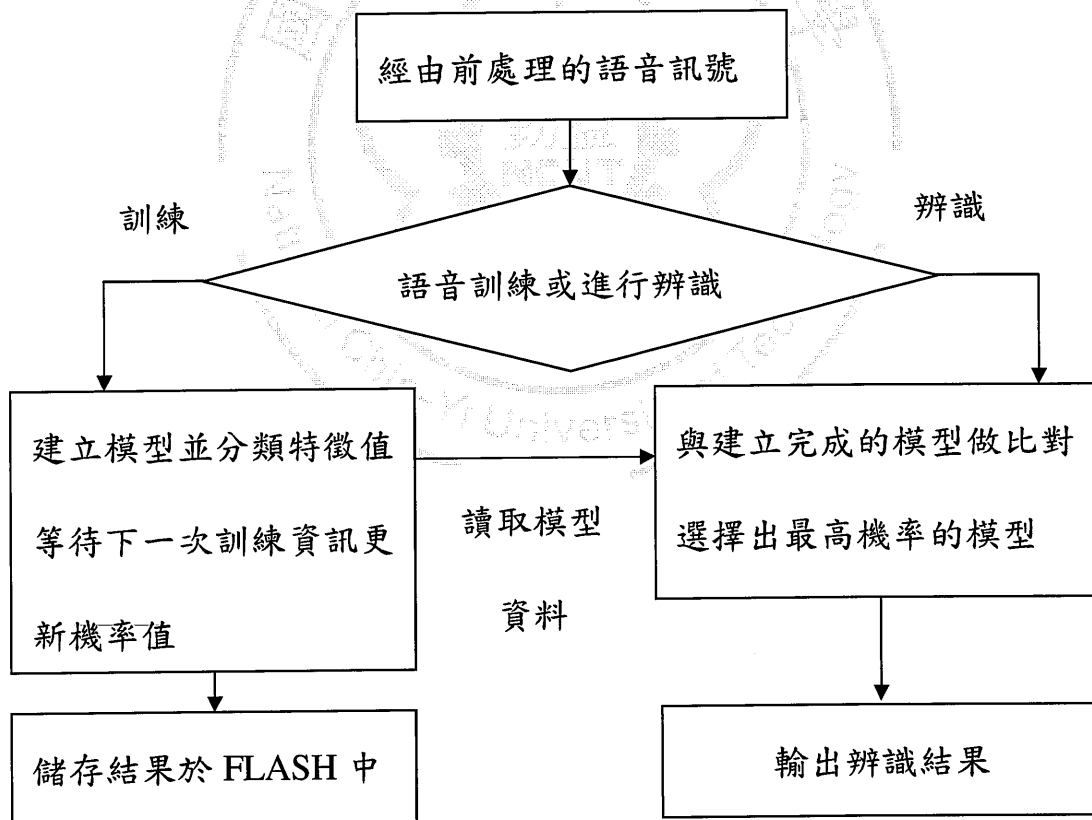


圖 4.2 語音辨識流程圖



## 4.1 隱藏式馬可夫模型的基本概念

首先本論文必須將隱藏式馬可夫模型的主要的元素做基本的說明：

隱藏式馬可夫模型參數可用  $\{A, B, \pi, S, V\}$  來表示，接下來將對每一個參數做解釋。

參數  $S$ ，代表有  $n$  個狀態， $S = \{s_1, s_2, \dots, s_n\}$ ，假設狀態數有 6 個，則  $S = \{s_1, s_2, s_3, s_4, s_5, s_6\}$ 。

參數  $V$ ，代表有  $m$  個產出結果， $V = \{v_1, v_2, \dots, v_m\}$ ，假設產出的結果有 10 個，則  $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}$ ，即為這些特徵值參數被定義的字碼。

參數  $A$ ，代表  $N \times M$  的狀態轉移機率矩陣，參數代表的意義如  $A = \{a_{ij}\}$ ,  $a_{ij} = P(q_t = s_j | q_{t-1} = s_i)$ ， $s_i$  為第  $i$  個狀態， $q_t$  為第  $t$  個時間的狀態。

參數  $B$ ，代表  $N \times M$  的觀察結果機率矩陣，參數代表的意義如

$B = \{b_j(k)\}$ ,  $b_j(k) = P(o_t = v_k | q_t = s_j)$ ， $v_k$  為出現字碼  $k$ 。

參數  $\pi$ ，為初始狀態機率， $\pi_i = \{\pi_1, \pi_2, \pi_3, \dots, \pi_i\}$ ，而  $\pi_i = P(q_1 = s_i)$ ，可以給入  $(1, 0, \dots, 0)$ ，因為一律從第一個狀態開始， $\pi_1$  的機率會是 1，其餘狀態的機率都是 0。

所以只要有  $N$  個單句，就建立  $N$  個馬可夫模型。我們給目前觀察狀態結果序列  $O = \{O_1, O_2, \dots, O_T\}$ 。

圖 4.3 為模組狀態序列， $T$  為序列長度， $n$  為狀態總數。而對語音而言，狀態的轉移只會由左向右，允許可以跳越一個狀態，或停留在原地。

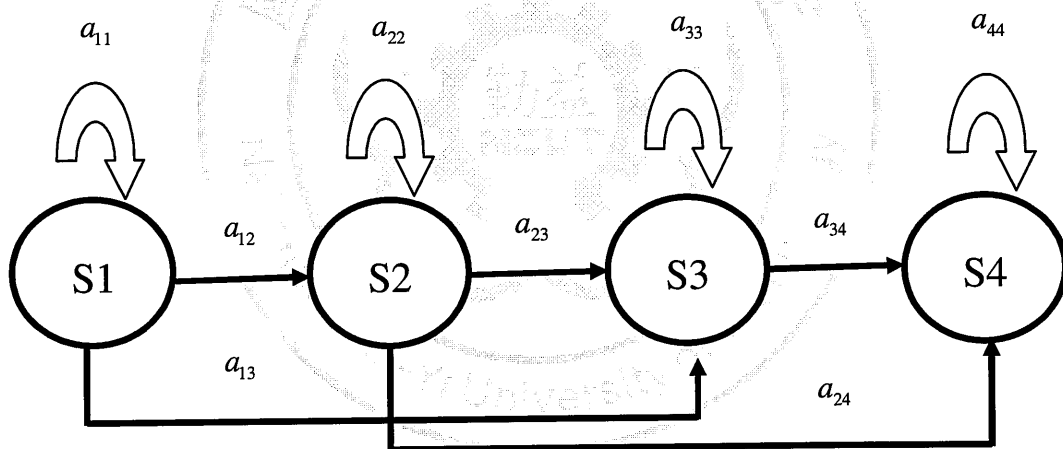


圖 4.3 狀態數為四的由左向右馬可夫鏈

我們先給定一個適當的狀態數量  $n$ 、狀態轉移機率  $A$ 、觀察結果機率  $B$ 、初始狀態機率  $\pi$ ，將  $O$  定義為觀察結果序列，也就是  $P(O|\lambda)$ ，但是  $O$  序列內的觀察結果皆為音框內的特徵值向量(已被定義為字碼)，而接下來我們就要敘述如何調整  $A, B, \pi$  這些參數，使  $P(O|\lambda)$  達到機率的

最大值。

我們可以簡單的對參數  $A, B, \pi$  作一些簡單的參數設定，例如說目前有 7 個狀態的音節組成一個單字，狀態中有 64 個觀察結果，因此可以對  $A$  的狀態轉移機率矩陣與  $B$  的觀察結果機率矩陣做一些設定。

表 4.1 隱藏式馬可夫模型  $\lambda$  狀態轉移機率矩陣  $A(7 \times 7)$

	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$
$S_1$	0.5	0.4	0.1	0	0	0	0
$S_2$	0	0.5	0.4	0.1	0	0	0
$S_3$	0	0	0.5	0.4	0.1	0	0
$S_4$	0	0	0	0.5	0.4	0.1	0
$S_5$	0	0	0	0	0.5	0.4	0.1
$S_6$	0.1	0	0	0	0	0.5	0.4
$S_7$	0.4	0.1	0	0	0	0	0.5

我們參考圖 4.3 再回來看表 4.1 的狀態模組可以看出一個規律，紅色框內部分相加起來的參數值正好為 1，也就是這些轉移機率相加起來為 100%。

表 4.2 隱藏式馬可夫模型  $\lambda$  觀察結果轉移機率矩陣  $B(7 \times 64)$

	$O_1$	$O_2$	$O_3$	$O_4$	$O_5$	...	$O_{64}$
$S_1$	0.005	0.004	0.003	0.003	0.002	...	0.0001
$S_2$	0.008	0.015	0.004	0.001	0.007	...	0.00001
$S_3$	0.003	0.022	0.004	0.004	0.004	...	0.00004
$S_4$	0.017	0.023	0.008	0.005	0.009	...	0.0007
$S_5$	0.019	0.018	0.01	0.008	0.006	...	0.0001
$S_6$	0.006	0.03	0.033	0.011	0.016	...	0.0004
$S_7$	0.04	0.007	0.008	0.008	0.003	...	0.0003

我們參考表 4.2 的結果，紅色框內相加起來的部分也必須正好為 1，也就是這些機率相加起來為 100%。

根據表 4.1、表 4.2 來看，簡單來說，這些狀態與狀態是互相連結的，  
 如圖 4.4 所示。假設狀態由  $S_1$  開始，路徑就有三個，延續下去。由上述  
 可知我們將  $\pi_1$  的機率設定為 1，所以一定只會從  $S_1$  開始。

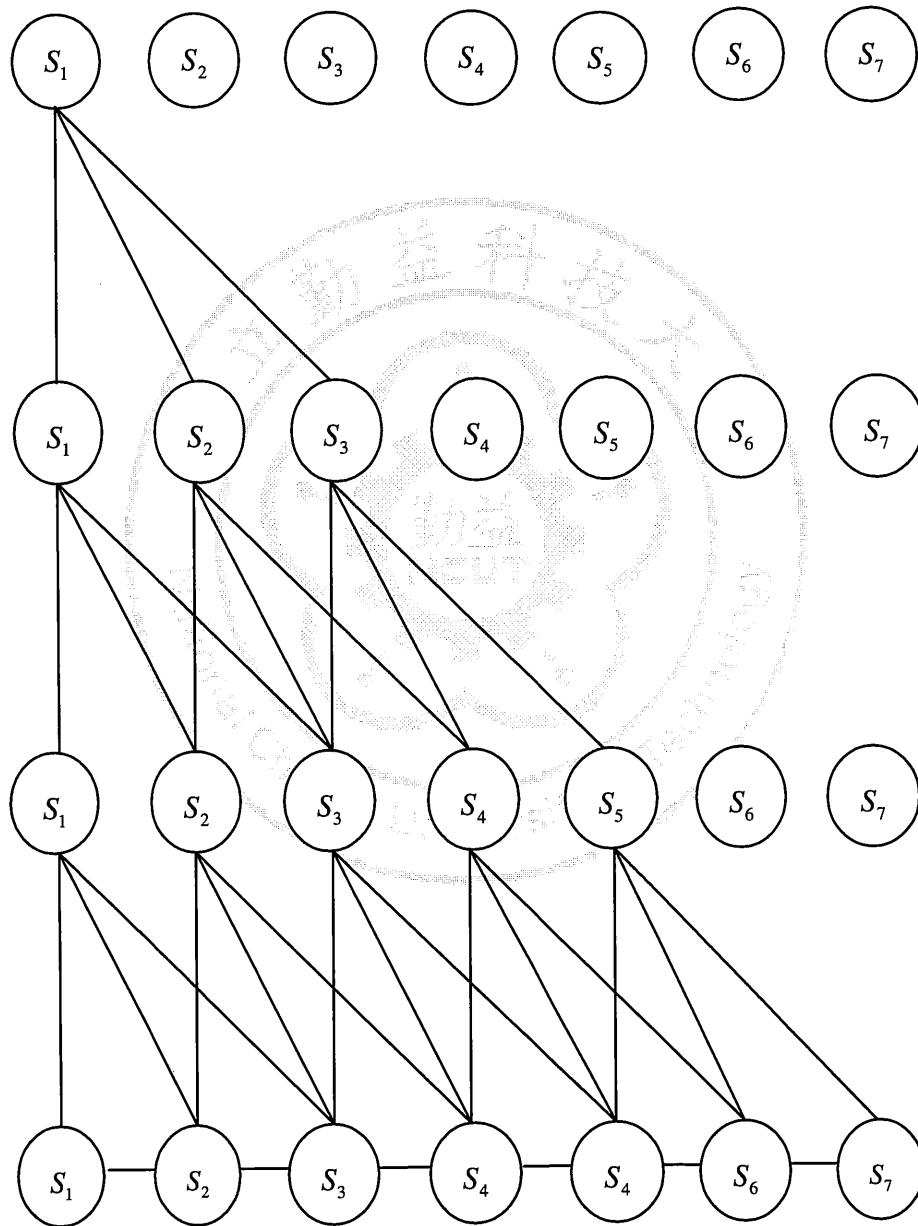


圖 4.4 狀態與狀態間之關係圖

## 4.2 隱藏式馬可夫模型產生結果機率的計算

在上述中提到假定一個隱藏式馬可夫模型為  $\lambda$ ，計算觀測結果序列為

$O = \{O_1, O_2, \dots, O_T\}$ ，最直接的方式就是計算  $n$  個狀態內所有機率的總

合，為了計算出狀態內機率的總合，我們定義一個狀態序列。

$Q = \{q_1, q_2, q_3, \dots, q_T\}$ ， $q_1$  是最初始的狀態，由以上的狀態序列  $Q$  中可以

得到觀察結果序列  $O$  的計算公式：

$$P(O|Q, \lambda) = \prod_{t=1}^T P(o_t | q_t, \lambda) = b_{q_1}(o_1) \cdot b_{q_2}(o_2) \dots b_{q_T}(o_T) \quad (4.1)$$

首先計算隱藏式馬可夫模型內各個狀態序列中的各個觀察結果，序列

由 1 至  $T$  個。

接下來產生選擇到模型  $\lambda$  中  $Q$  這個狀態序列的機率值

$$P(Q|\lambda) = \pi_{q_1} \cdot a_{q_1q_2} \cdot a_{q_2q_3} \dots a_{q_{T-1}q_T} \quad (4.2)$$

由以上的兩個公式可以看出選擇到狀態序列  $Q$  及觀察結果  $O$  的機率，

接下來我們把這兩個公式相成，取得一個聯合機率值。

$$P(O, Q|\lambda) = P(O|Q, \lambda)P(Q|\lambda) \quad (4.3)$$

將所有的參數帶入公式，也就是將狀態序列與觀察結果機率值相乘得

$$\sum_{all Q} P(O, Q|\lambda) = \sum_{q_1, q_2, \dots, q_T} \pi_{q_1} \cdot b_{q_1}(o_1) \cdot a_{q_1q_2} \cdot b_{q_2}(o_2) \cdot a_{q_2q_3} \dots a_{q_{T-1}q_T} \cdot b_{q_T}(o_T) \quad (4.4)$$

$\pi_{q_1}$  代表最初始的狀態機率， $b_{q_1}(o_1)$  代表第 1 個狀態下在觀察結果機率矩陣中找到的觀察結果機率， $a_{q_1q_2}$  代表第 1 個狀態轉移至第 2 個狀態的機率，可在狀態機率矩陣  $A$  中取得。在以上的理論之中我們的是可以計算  $P(O|\lambda)$ ，在是在某種情況下計算的量會非常大，在這裡舉一個簡單的例子，在模型  $\lambda$  中有  $n$  個狀態在序列  $Q$  中，而觀測序列中有  $T$  個觀測值，則所有狀態轉換序列數目就為  $N^T$ ，根據公式(4.4)的結果顯示需要  $O(2TN^T)$  的乘法運算量，即使有 4 個狀態數，時間長度為 25 個單位(ms)，那麼以乘法的計算來看就有約  $5.52 \times 10^{16}$ ，這對於嵌入式系統是一個非常大的計算量，就算把演算出來的結果套用，嵌入式系統的暫存器也不能容納。所以我們將公式(4.4)做改變，因此一般來說都會使用向前-向後程序的演算法來計算隱藏式馬可夫模型的機率值。

### 4.3 向前-向後演算法

本論文中的上一章節提到要如何簡化計算的乘法，先前我們已經給定某一個模型參數  $\lambda$ ，在這裡我們再給定一個向前的變數如公式(4.5)以方便作小量範圍的演算，其示意圖如圖 4.5 所示。

$$\alpha_t(i) = P(o_1, o_2, \dots, o_t, q_t = s_i | \lambda) \quad (4.5)$$

$\alpha_t(i)$  表示出現觀測序列中  $\{o_1, o_2, \dots, o_t\}$  在  $t$  的時間內，停在狀態  $s_i$  的機率。

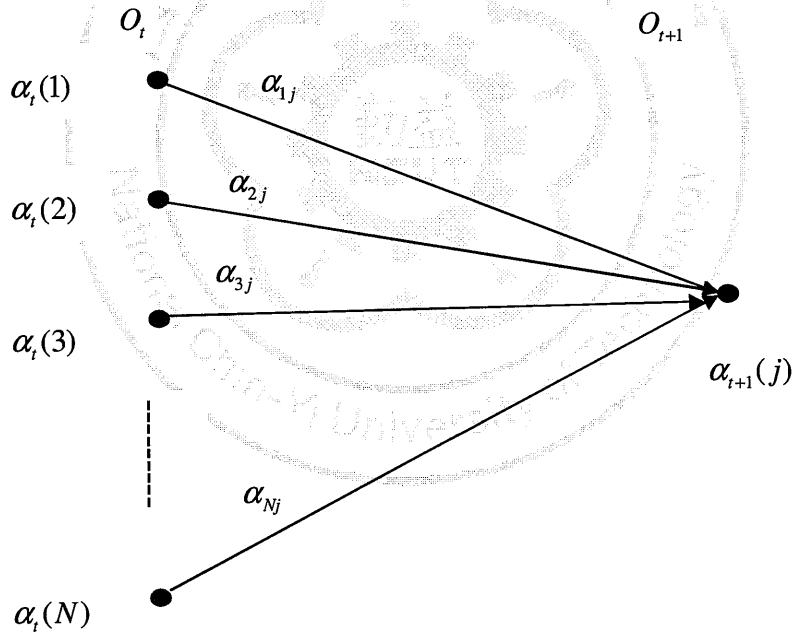


圖 4.5 向前演算法示意圖



### 4.3.1 向前演算法之初始化

首先我們先給向前變數一個初始值

$$\alpha_1(i) = \pi_i b_i(o_1) \quad (4.6)$$

$\alpha_1(i)$  代表初始狀態時  $s_i$  停留的機率值。

### 4.3.2 向前演算法之遞迴

將這些參數逐一套入公式(4.6)中，並進行演算。

$$\alpha_{t+1}(j) = \left[ \sum_{i=1}^N \alpha_t(i) \cdot a_{ij} \right] \cdot b_j(o_{t+1}), 1 \leq j \leq N, t = 1, 2, \dots, T-1 \quad (4.7)$$

$\alpha_{t+1}(j)$  代表下一個時間狀態  $s_j$  的機率值， $\alpha_t(i)$  代表上一個狀態  $s_i$  的機率值， $b_j(o_{t+1})$  代表在狀態  $s_j$  的觀察結果機率。

### 4.3.3 向前演算法之結束

將所有的  $s_i$  套入運算出模型內觀察結果的機率值。

$$\begin{aligned} P(O | \lambda) &= \sum_{i=1}^N \alpha_T(i) \\ &= \sum_{i=1}^N \alpha_t(i) \beta(i) \\ &= \sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j) \end{aligned} \quad (4.8)$$

$\beta(i)$  代表在狀態  $s_i$  會出現  $\{o_{t+1}, o_{t+2}, \dots, o_T\}$  的觀察結果機率。  $\beta_{t+1}(j)$  代表在  $t+1$  時間狀態  $s_j$  的觀察結果機率值。運用此向前演算法能夠有效的減少乘法數量，舉一個簡單的例子來說，若與先前一樣有 4 個狀態，時間長度為 25 個單位(ms)，則乘法的數量會減少為  $(T-1)N^2 + N$  個，也就是說會減少至 388 個，比公式(4.4)的  $5.52 \times 10^{16}$  計算量少得太多了。而為了之後找出狀態中移動的所有狀態機率值，必須還要定義一個向後演算法。

#### 4.4 向後演算法

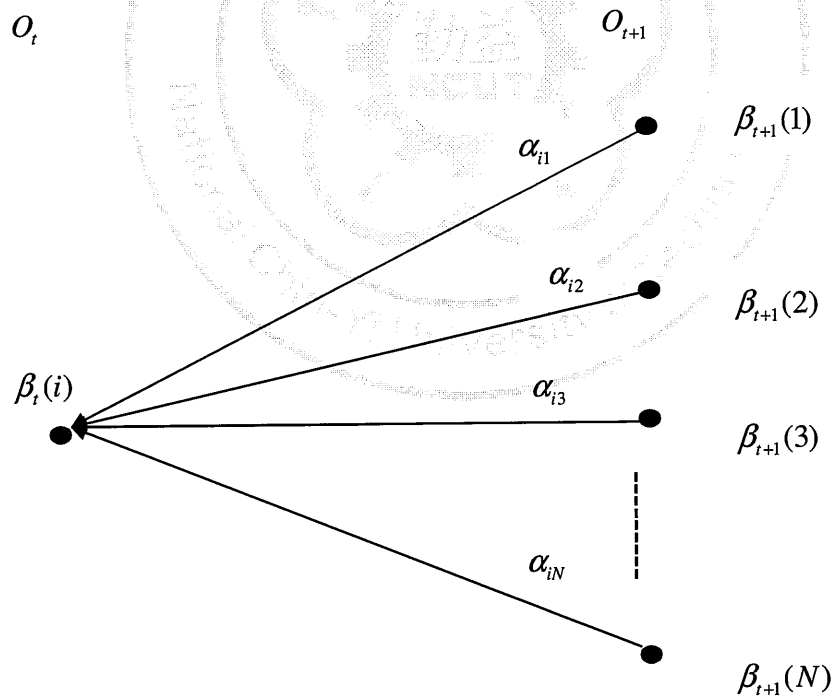


圖 4.6 向後演算法示意圖

程序上與先前的算法類似，其示意圖如圖 4.6 所示。

#### 4.4.1 向後演算法之初始化

$$\beta_T(i) = 1, 1 \leq i \leq N \quad (4.9)$$

$\beta_T(i)$  代表最初始機率值，指的是在  $T$  時間狀態  $s_i$  機率的總和為 1， $n$  代表所有狀態的數量。

#### 4.4.2 向後演算法之遞迴

$$\beta_t(i) = \sum_{j=1}^N a_{ij} \cdot b_j(o_{t+1}) \cdot \beta_{t+1}(j), 1 \leq i \leq N, t = T-1, T-2, \dots, 1 \quad (4.10)$$

以向前-向後演算法則可以計算出停留在  $s_i$  狀態的機率值及  $s_i$  狀態轉移出去的機率值，同時也是為了方便之後參數重估訓練的部分。

## 4.5 隱藏式馬可夫模型的參數估算

以上的參數幾乎都是未知的，大部分要藉由訓練不斷的更新參數，將基本的  $A, B, \pi$  求出，但是我們必須還要修正狀態轉移及觀察機率矩陣的機率，使的  $P(O|\lambda)$  估算機率為最大值，也就是說，在即時語音辨識中，需要藉由訓練後的語音樣本去修正隱藏式馬可夫模型。

與上述一樣，先給定模型一個參數  $\lambda$ ，與觀測序列  $O$ ，再來需要定義一個機率變數  $\zeta_t(i, j)$ ，此參數被定義為從時間  $t$  到  $t+1$  時，狀態由  $s_i$  轉移到  $s_j$  的機率，可以用公式(4.11)來表達。

$$\zeta_t(i, j) = \frac{P(q_t = s_i, q_{t+1} = s_j, O | \lambda)}{P(O | \lambda)} \quad (4.11)$$

接下來本論文將先前提到的向前-向後演算程序之參數帶入公式

$$\zeta_t(i, j) = \frac{\alpha_t a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O | \lambda)} \quad (4.12)$$

而我們能夠知道公式(4.12)的分母為：

$$P(O | \lambda) = \sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$$

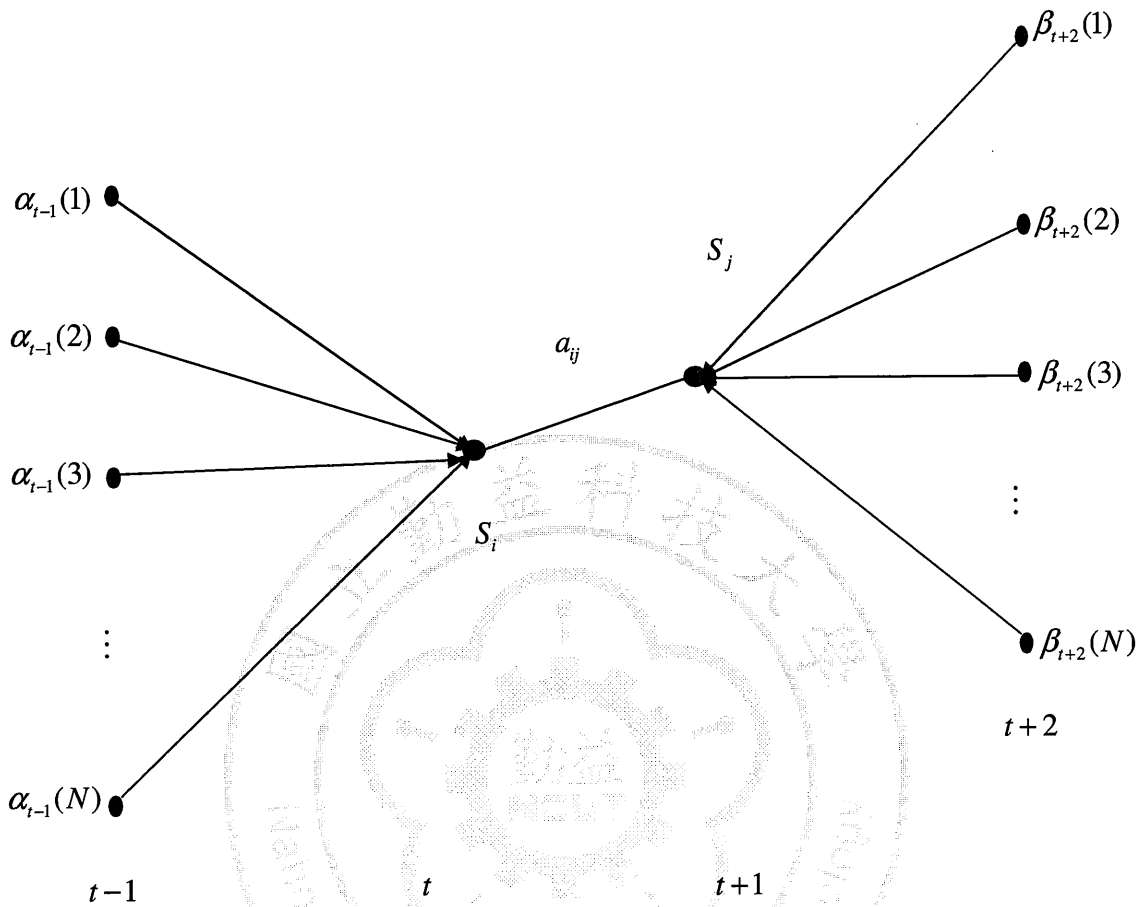


圖 4.7 狀態機率轉移演算示意圖

由圖 4.7 能夠知道在  $t$  時間之下，在狀態  $S_i$  的機率值。接下來就要重估狀態轉移機率值。

在這裡先定義一個參數  $\gamma_t(i)$ ，初始化的第一個時間內  $\bar{\pi}_i = \gamma_1(i)$  而

$$\gamma_t(i) = \sum_{j=1}^N \zeta_t(i, j) \quad (4.13)$$

$\gamma_t(i)$  代表  $t$  時間內  $s_i$  停留在自己狀態的機率。

如圖 4.7 中，在一整個  $T$  個時間中，有  $t-1$  次的機會是從狀態  $S_i$  出現的，

其中有一部分會進入  $S_j$ ，因此可以估算狀態  $S_i$  進入  $S_j$  的轉移機率：

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \zeta_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad (4.14)$$

$\bar{a}_{ij}$  為狀態轉移機率的重估值，公式(4.14)的分母部分為停在狀態  $S_i$  的參數值，分子部分為狀態  $S_i$  轉移到  $S_j$  的參數值。

接下來我們還要對狀態的觀察結果機率進行重估：

$$\bar{b}_j(k) = \frac{\sum_{t=1}^T \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)} \quad (4.15)$$

公式(4.15)之  $\bar{b}_j(k)$  為狀態觀察結果機率的重估值，分母部分是停留在狀

態  $S_j$  的參數值，分子部分是所有停留在  $S_j$  時有輸出觀測結果  $v_k$  的參數值。以上的估算方式叫做維特比演算法，實際上的做法就是對一個辨認單位的模型，拿所有這個辨認單位的訓練語音來帶入維特比演算法得到一群最佳狀態序列，從這些最佳狀態序列中統計狀態轉移的次數，以及產生觀察結果的次數，就可以利用重估的方式更新參數，再以新的參數更新模型，如此重複讓模型參數更新，最後就能得到辨認單位的模型。

#### 4.6 語音辨識的訓練演及分類算法

語音辨識的訓練方式有許多種，在這裡利用 k-mean 樣本分類的方式，也就是將這些需要受訓練的語音特徵值向量進行分類。在此將這些特徵值向量進行量化，再來將訓練中相同性質的部分歸為一類，而類別的元素就是這些被量化的特徵值。但是我們知道這些元素之間一定存在著某個程度的誤差，若將全部欲分類之元素的個別誤差累計起來就是整體的個別誤差，所以只要能找到最小的個別的類別誤差，就能夠解決樣本分類的問題了。而這些誤差當然是越小越好，所以我們必須不斷反覆的進行分類，將整體的誤差降到最低，最後到達收斂。

#### 4.6.1 訓練資料

首先將語音資料經由麥克風之後進行音框切割等工作，完成特徵值擷取與量化的條件。

#### 4.6.2 初始化中心值

對每一個類別一開始給定一個初始的中心值，可以將第一次訓練的量化參數做為中心值，由於之後還會在進行更新，所以這個步驟是暫時的。

#### 4.6.3 對元素進行分類

將每一次進行訓練的資料(同一個音)放入類別中，並且與初始化的參數進行比較，找出誤差較小的元素，而其他誤差較大的元素就是訓練失敗的部分，可以不列入類別中，也可暫時保留，等待下一步驟更新中心值參數時提取出來。

#### 4.6.4 更新類別的中心值

將這些誤差較小的元素進行中心值的計算，也就是算出該類別的中心值。簡單來說就是將各個元素相加過後除以該類別的元素個數來計算中心值。



#### 4.6.5 判斷收斂與否

排除那些差異較大的元素，可再經由訓練的資料來進行中心值的更新，而這些中心值參數通常與初始化選取的中心值有關。當這些元素與中心值的誤差到達一個較小的差別時，就代表收斂了，而中心值與元素之間的誤差門檻可自行設定，但是通常訓練的次數會超過 1000 次以上。

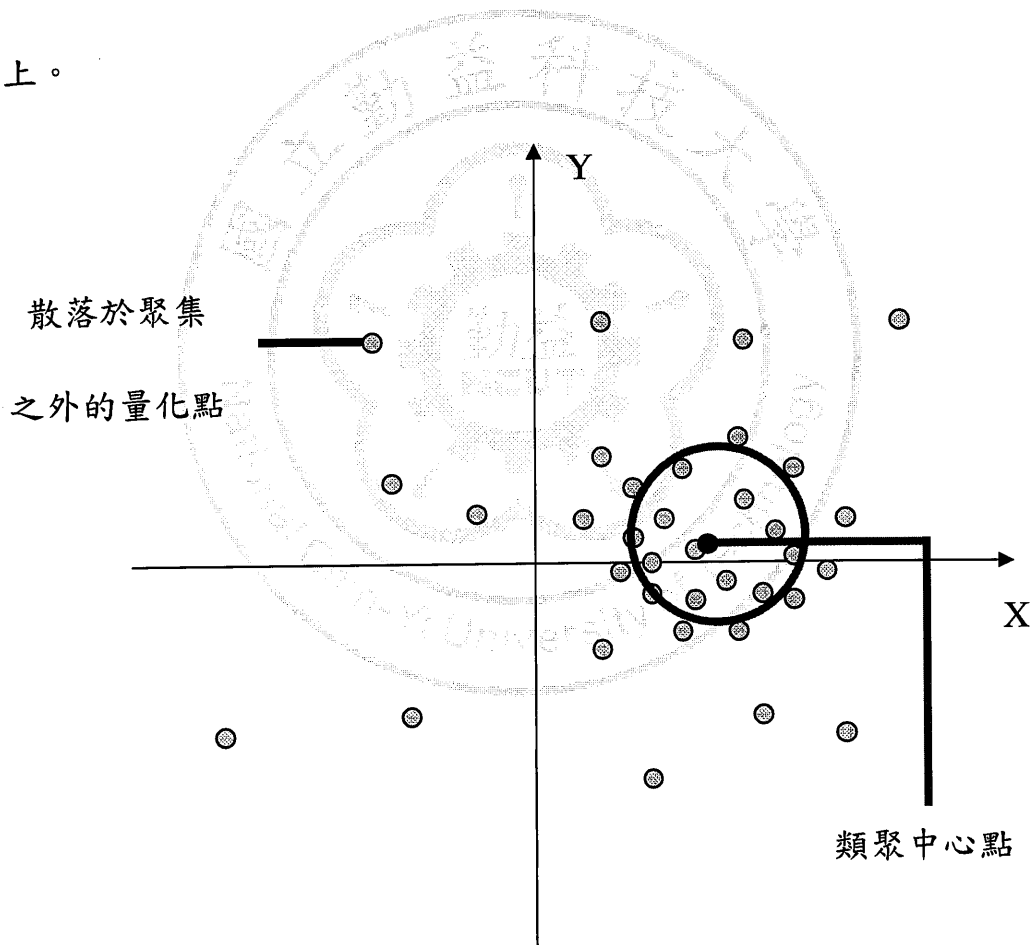


圖 4.8 k-mean 於語音訓練結果之類聚圖

我們找到這些類聚中心點之後就可以對這些中心點，也就是說這些中心點就是觀察結果序列。利用這些中心點去設定相對的字碼，而我們就可以利用這些相對應的字碼去更新並計算隱藏式馬可夫模型  $a_{ij}$ 、 $b_j$ 、 $\pi_i$  的機率值，簡單來說就是利用參數估算去更新機率值。

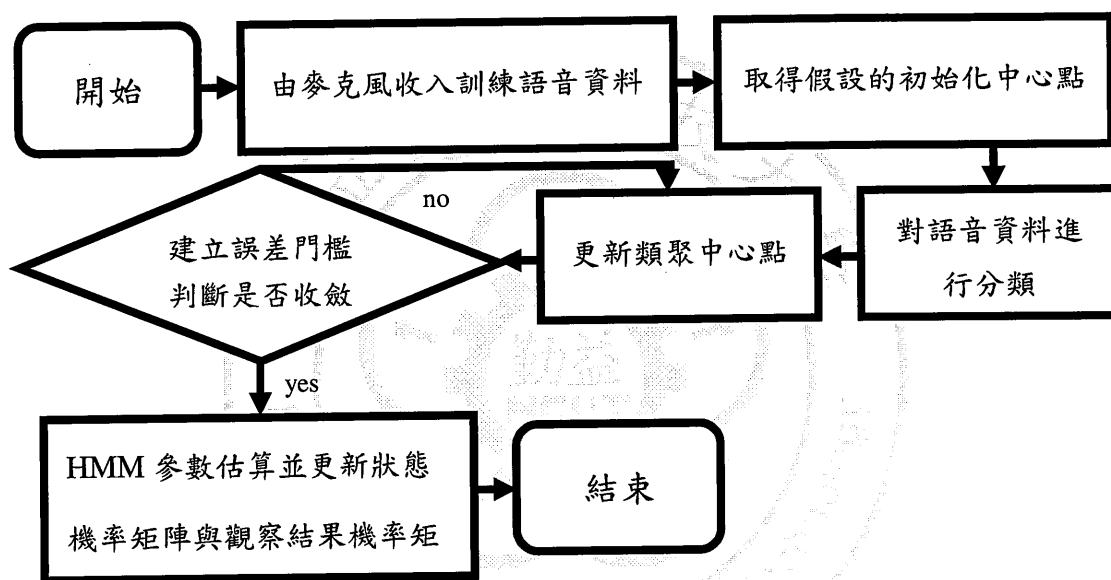


圖 4.9 訓練演算法之流程圖

## 4.7 隱藏式馬可夫模型最佳狀態序列的計算

當我們給定一個模型  $\lambda$ ，導入適當的  $A, B, \pi$  後，就必須去修正隱藏式馬可夫模型上的狀態機率，使其成為最大值。可以從圖 4.1 之中看出在  $n$  個狀態與  $T$  個時間下，從左到右的任何一條路徑就有能是一個狀態序列。如果  $O = \{O_1, O_2, \dots, O_T\}$  這個觀察的結果是已知的，那麼對應的這個觀察結果序列應該就有一個最佳的狀態序列能夠追尋。以下的演算過程就是要找出一條由左到右的最佳路徑。

由以上的向前-向後演算法可以知道：

$$P(O, q_t = s_i | \lambda) = \alpha_t(t) \beta_t(i) \quad (4.16)$$

而我們在這裡將  $P(O, q_t = s_i | \lambda)$  定義為  $\gamma_t(i)$

也就是說  $\gamma_t(i) = P(O, q_t = s_i | \lambda)$ ，而在時間點  $t$  我們必須要找到一個適當的  $s_i$  使  $\gamma_t(i)$  的機率為最大，可以用一個簡單的公式來定義：

$$q_t^* = \arg \max_{1 \leq i \leq N} \{ \gamma_t(i) \} \quad (4.17)$$

最後將  $q_t^*$  串連起來就是一條最佳狀態序列，而在二維平面看到的就是一條完整的最佳路徑，示意圖就如同圖 4.10 所示，而在此就必須要用到維特比演算法。

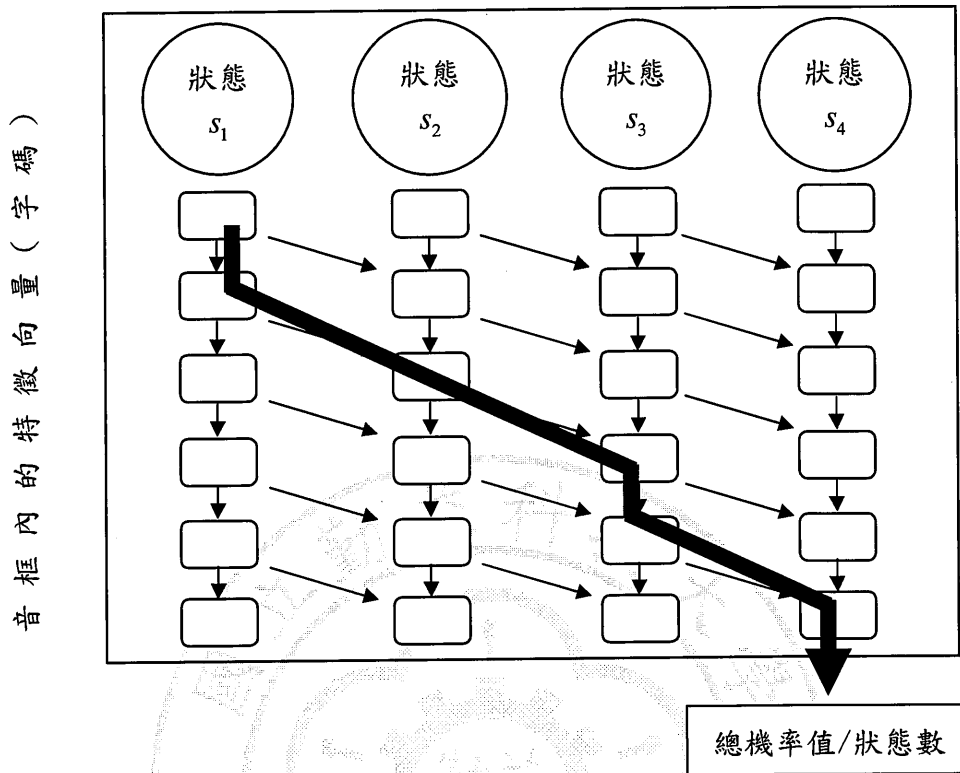


圖 4.10 狀態與假設音框關係圖

現在我們必須要建立一個新的變數  $\delta_t(i)$ ，此參數用來表達在隱藏式馬可夫模型  $\lambda$  中，狀態  $i$  中的所有最大機率的觀察結果之機率值，可由公式(4.18)來表示：

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P(q_1, q_2, \dots, q_{t-1}, q_t = s_i, o_1, o_2, \dots, o_t | \lambda) \quad (4.18)$$

參數  $\delta_t(i)$  主要用來表示在  $t$  時間時進入狀態  $s_i$  出現的最大值。

$O = \{O_1, O_2, \dots, O_T\}$ ，而在  $t$  時間的狀態為  $s_i$  的最大機率，由以上的解釋我們可以得到另外一個公式(4.19)來表達，取得下一個時間狀態的最大機率值。

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] b_j(o_{t+1}) \quad (4.19)$$

$\delta_{t+1}(j)$  代表在  $t+1$  時間時狀態  $s_j$  的最大機率值。

最後我們在定義一個變數  $\psi_t(i)$  來記錄選取的狀態，接下來本篇論文利用維特比演算法(Viterbi algorithm)算出此語音在整個模組中的最佳狀態序列機率，由於他只考慮前一級的最佳狀態並可記錄下累積路徑最大機率值，可以節省計算時間成本。

而維特比主要的演算步驟如下：

#### 4.7.1 維特比演算法初始化

$$\left\{ \begin{array}{l} \delta_1(i) = \pi_i \cdot b_i(o_1) \\ \psi_1(i) = 0 \end{array} \right\}, 1 \leq i \leq N \quad (4.20)$$

$\psi_t(i)$  是用來記錄選取的狀態。

#### 4.7.2 維特比演算法之遞迴

$$\left\{ \begin{array}{l} \delta_{t+1}(j) = \max_{1 \leq i \leq N} [\delta_t(i) a_{ij}] b_j(o_{t+1}) \\ \psi_{t+1}(j) = \arg \max_{1 \leq i \leq N} [\delta_t(i) a_{ij}] \end{array} \right\}, 1 \leq j \leq N, 1 \leq t \leq T-1 \quad (4.21)$$

$\delta_{t+1}(j)$  為  $t+1$  時刻狀態  $j$  的機率,  $\psi_{t+1}(j)$  為  $t+1$  時刻到達狀態  $j$  的機率。

#### 4.7.3 維特比演算法之結束

$$\left\{ \begin{array}{l} P^* = \max_{1 \leq i \leq N} [\delta_T(i)] \\ q_T^* = \arg \max_{1 \leq i \leq N} [\delta_T(i)] \end{array} \right\} \quad (4.22)$$

$P^*$  代表計算出來的許多個  $s_i$  乘上觀察結果機率的值, 所以用  $*$  來表示此參數為陣列。

#### 4.7.4 維特比演算法之追溯路徑

$$q_t^* = \psi_{t+1}(q_{t+1}^*), t = T-1, T-2, \dots, 1 \quad (4.23)$$

$q_t^*$  代表將最大的  $s_i$  帶入所記錄的參數中乘上狀態轉移機率  $a_{ij}$  的結果。

根據  $\psi_{t+1}(j)$  的紀錄, 進行反向路徑找出最佳狀態序列。最後的追溯路徑則是將所有最大狀態機率與狀態內的觀察結果機率相乘放入  $q_t^*$  中, 並且將這些狀態串接起來。

## 第五章 聲音的預錄及播放

首先本論文收錄了許多不同的聲音，分別是菜單名稱及數量單位名稱，將這些取得的聲音儲存成 MP3 的格式，以節省儲存空間，播放時也不至於過度失真。

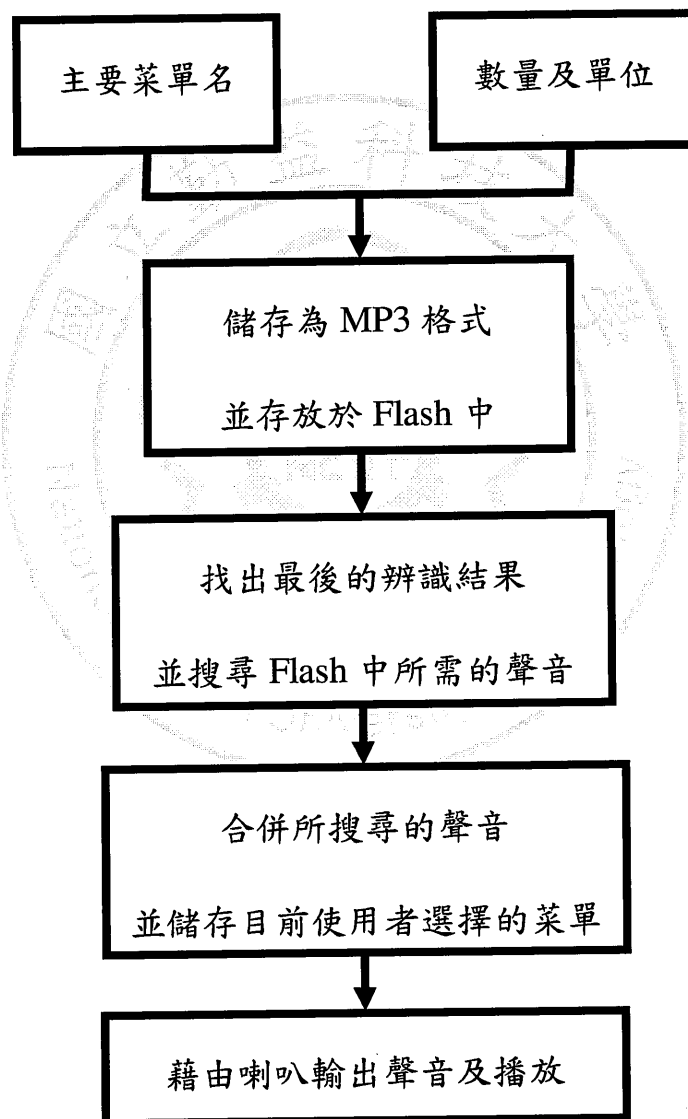


圖 5.1 聲音的預錄及輸出方塊圖

由以上的演算法中計算機率值並輸出結果選擇 Flash 中所需要的聲音。我們將這些選擇過後的聲音做整合並播放，增加機器人與人類互動的機會，並且可以確認使用者所選擇的餐點是否正確。本論文在這裡舉一個簡單輸出的聲音例子，您選擇了“豬排飯”“1”份，豬排飯為菜單名稱，1 份為數量名稱，而其他的文字則是固定的，可以另外再 Flash 中儲存，做更多變化。





## 第六章 實驗結果

本篇論文將以上方法在 SPCE3200 中實現，如圖所示。我們將各個菜單的選擇結果顯示於 SPCE3200 的 TFT 觸控螢幕中顯示，以便開發者方便操作，最後將選擇後的結果藉由串列傳輸與智慧型點餐服務機器人互相連結溝通秀出在觸控式螢幕給使用者參考決定。圖 6.1 中在菜單右邊的白圓是使用者對麥克風說話後系統辨識出來的餐點。

圖 6.1 菜單的內容有義大利麵、海鮮焗烤飯、日式咖哩飯、豬排飯，而最後的湯品則是轉移到下一個餐點的識別句。

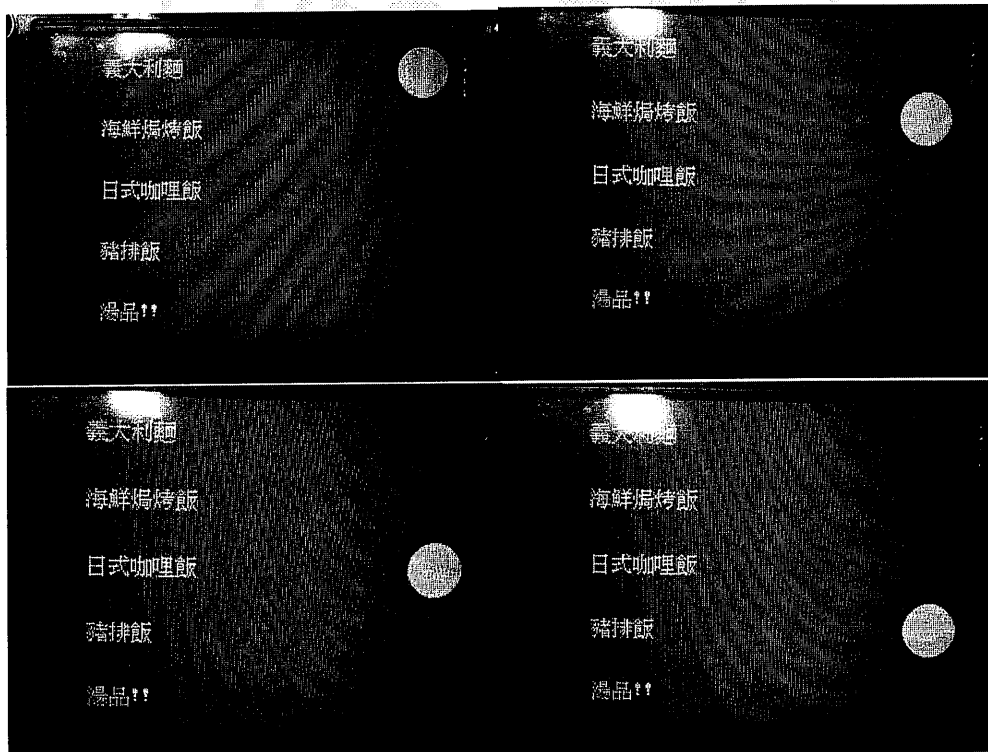


圖 6.1 實驗結果之菜單主餐圖

圖 6.2 主要是湯品的部分，內容有魚丸湯、貢丸湯、排骨湯、玉米濃湯，最後的飲料是要轉移到下一個餐點的識別句。

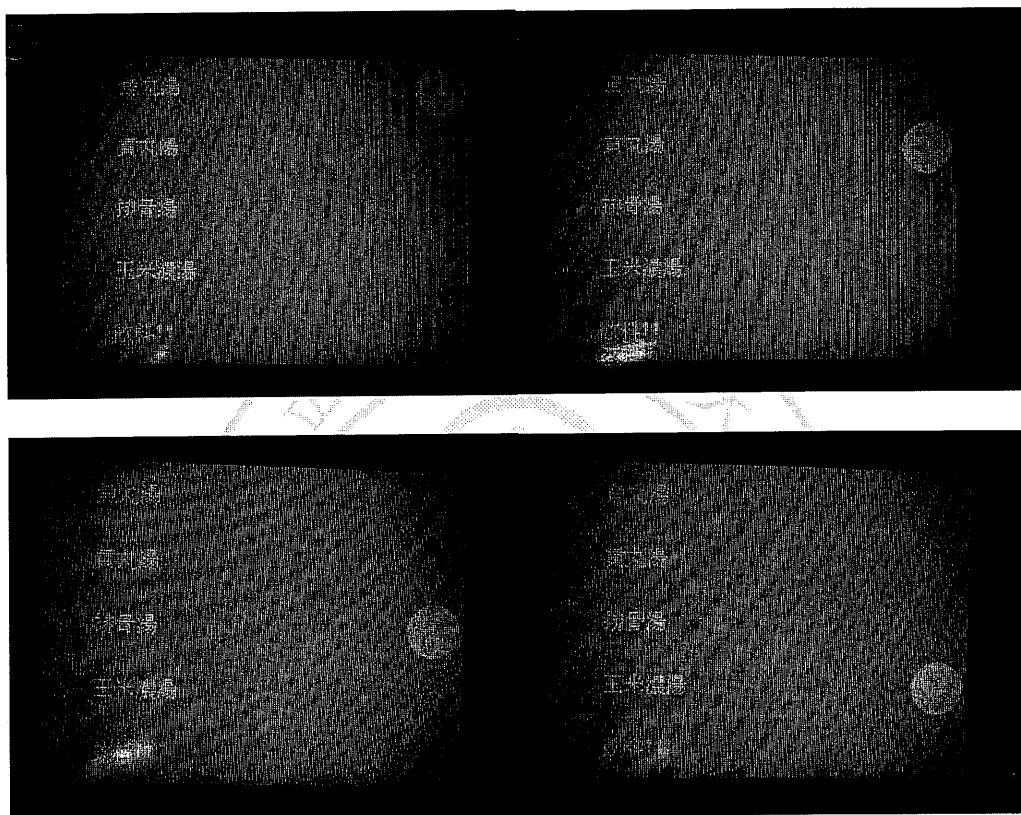


圖 6.2 實驗結果之菜單湯品圖

圖 6.3 主要顯示的是飲料的部分，內容有紅茶、綠茶、柳橙汁、麥茶。

最後的點心是要轉移到下一個餐點的識別句。

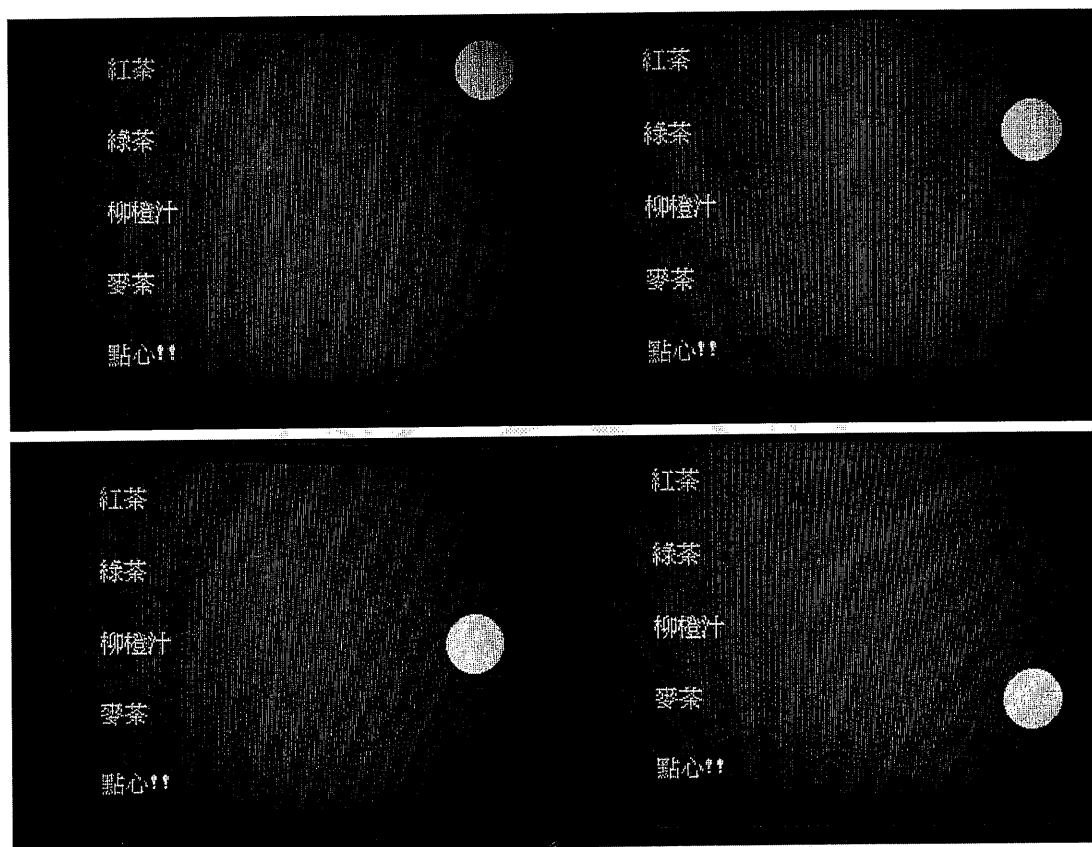


圖 6.3 實驗結果之菜單飲料圖

圖 6.4 主要顯示的是點心的部分，內容有提拉米蘇、冰淇淋、仙草凍、巧克力藍莓。最後的水果是要轉移到下一個餐點的識別句。

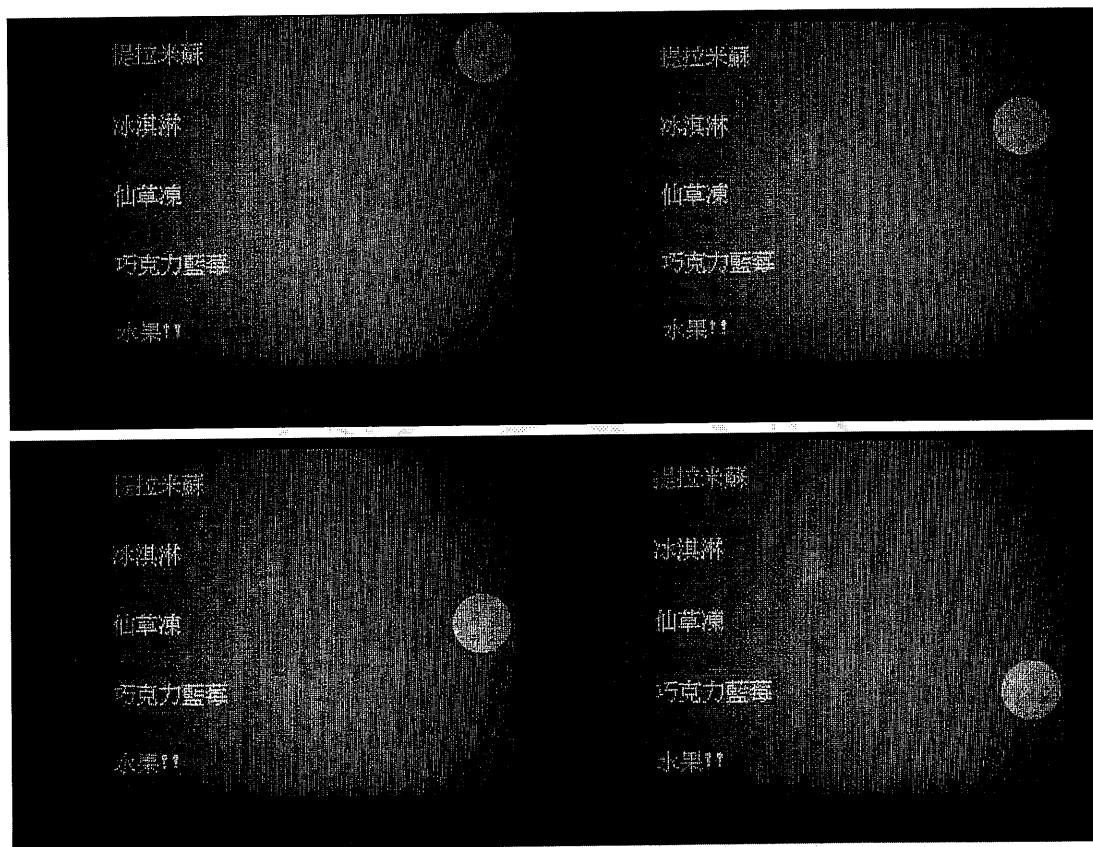


圖 6.4 實驗結果之菜單水果圖

圖 6.5 主要顯示的是水果的部分，內容有西瓜、草莓、水蜜桃、蘋果。最後的識別句完成是為了使用者點完餐點後所設計的，代表已經點完所有的餐點，系統會整理並說出使用者所點的菜單做參考，並做結帳的動作。

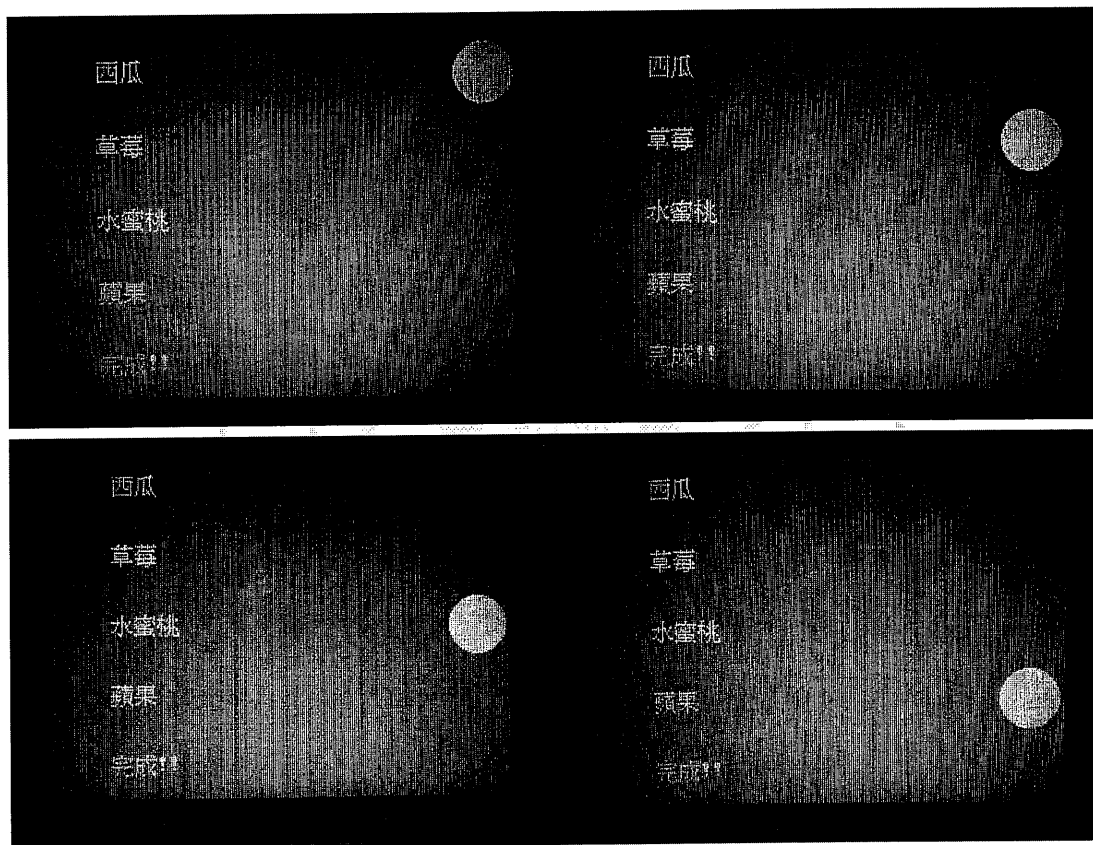


圖 6.5 實驗結果之菜單水果圖

本篇論文將 20 組中文餐點語音加入訓練，取樣頻率採用 16kHz，實驗於三種不同的環境下，每個環境下測試每組個語音資料 50 次 ( $3 \times 20 \times 50$ )，總共有 3000 筆的資料。第一個測試場地於勤益科技大學工程館 6 樓，約  $10\text{m} \times 10\text{m}$  的開放空曠場地。場地二測試於空間較場地三大且有回音，大約在  $10\text{m} \times 8\text{m}$  封閉空間。場地三測試於封閉狹窄的  $8\text{m} \times 2\text{m}$  小空間。最後加入預錄的聲音做整合的測試結果，一樣在三個不同的環境中每次測試 50 次，並計算平均的辨識率，統整所有菜單的訓練次數。圖 6.6 為第一個測試的場地辨識率，由於在戶外測試的環境因素影響較多，經常會造成雜訊干擾以，同時也能明顯觀察到辨識率較低的問題，圖 6.7 為第二個測試場地的辨識率，在室內環境中相對較安靜，但仍然有回音的干擾，導致無法識別，但相較於室外環境的辨識成功率卻有大幅的提升。圖 6.8 為第三個測試場地的辨識率，場地三於封閉的小房間中，無噪音以及回聲的干擾，與先前的兩個場地相比平均辨識率高了許多。以上的三個測試場地都是對所有的餐點做辨識，圖 6.9 代表每一個餐點的訓練次數，而本論文訓練的場地是在無干擾與雜訊的環境下訓練的，最後圖 6.10 代表西瓜的前 20ms 的特徵值參數，取特徵數量為 1000 個。

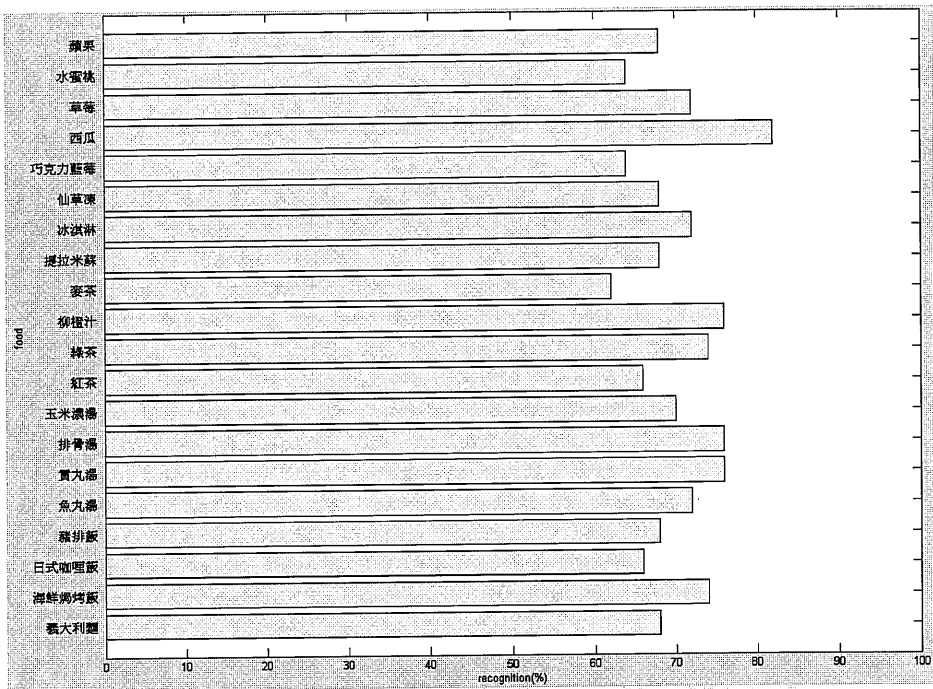


圖 6.6 於第一個實驗場地之辨識率

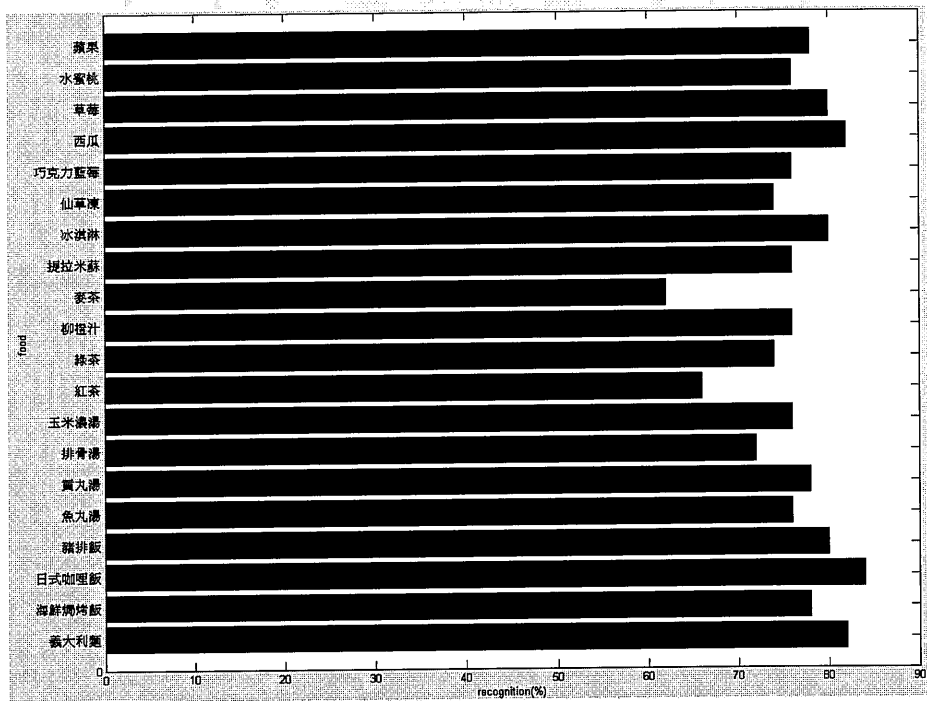


圖 6.7 於第二個實驗場地之辨識率

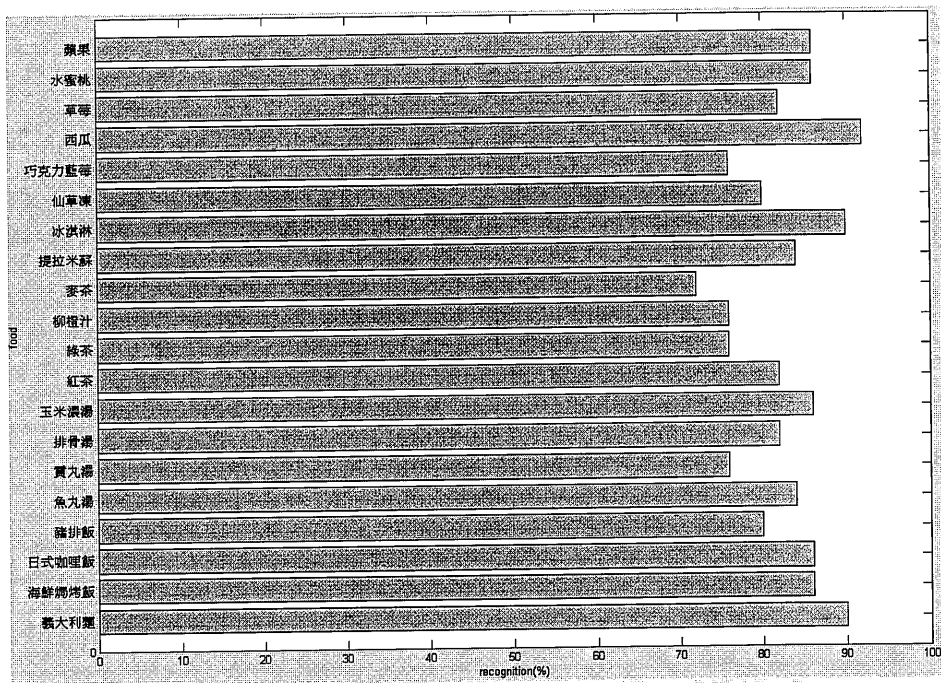


圖 6.8 於第三個實驗場地之辨識率

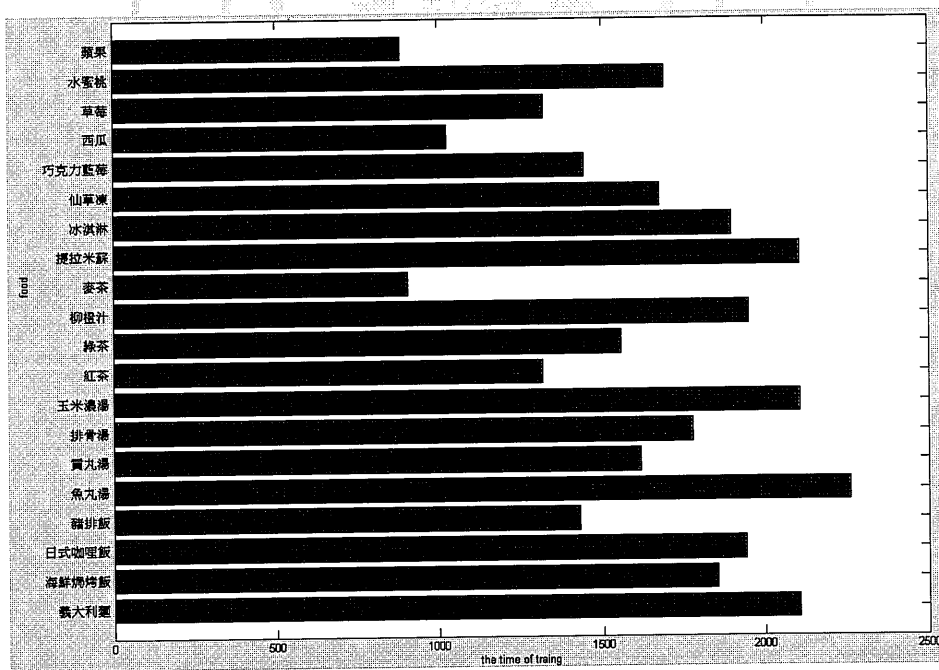


圖 6.9 總訓練次數直方統計圖



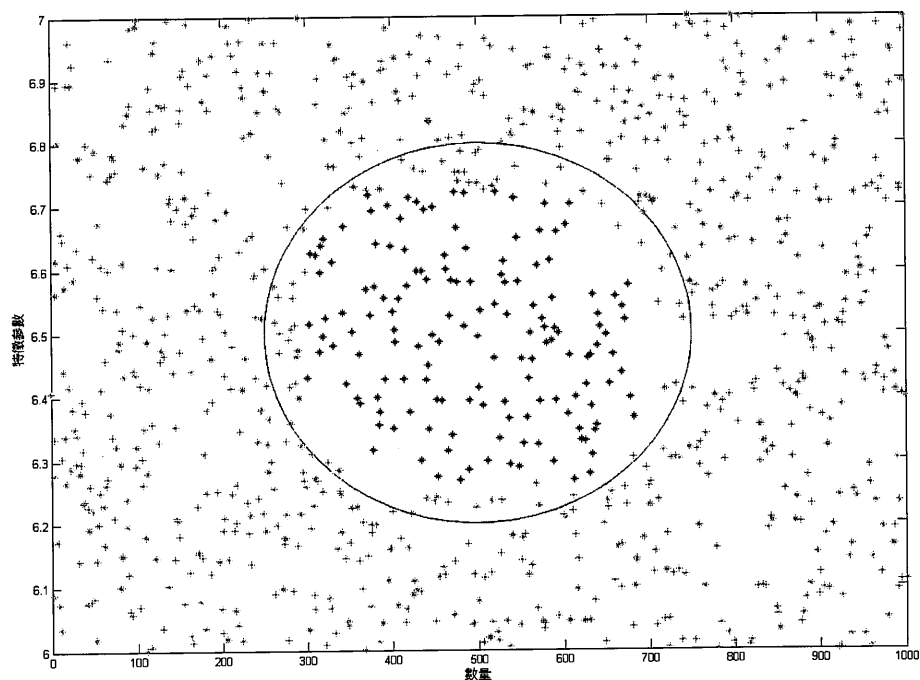


圖 6.10 語音特徵值收斂範圍

圖 6.6 測試場地一的平均辨識率為 70.3%，辨識率最高的文字為西瓜，辨識率達 82%，而辨識率最低的文字為麥茶 62%，原因在於在較差的環境下辨識麥茶時大部分都會辨識到紅茶，這讓模型中的觀察結果被多次誤判為麥茶的字碼，而這與訓練次數有相對的關係。圖 6.7 場地二的平均辨識率為 76.3%，辨識率最高的文字為日式咖哩飯 84%，辨識率最低的文字一樣為麥茶 62%，麥茶在 7 個狀態之中，每一個音節佔據的字碼大概有 15 個左右，這也表示它容易辨識到相同時間字句，觀

察到的字碼有部分的範圍是相同的，而誤判的字句一樣在紅茶。圖 6.8 場地三的平均辨識率為 82.6%，此方法的辨識率與傳統的 DTW[19]相比較為穩定。此方法中辨識率最高的文字為西瓜 92%，辨識率最低的文字為麥茶 72%，這就能確定必須將麥茶的字碼與紅茶的字碼做區別，盡量將兩者的類聚的中心點做區別。圖 6.9 為每一個菜單的訓練次數，可以看出文字較短的部分，訓練次數較少，且每次訓練的時間若都差不多，則特徵值較容易收斂，相對的辨識率也較高，而在訓練的部分，總平均訓練的次數為 1644 次，最後圖 6.10 是語音特徵值的收斂範圍，設定的門檻誤差值大約在 10%左右，紅色點為中心值，黑色點為誤差範圍內的特徵值，綠色點為訓練失敗或散落的特徵點，而門檻值設定在 10%的意義在於能夠較快速的完成特徵值統整及訓練，同時也可對這些散落的特徵值做類聚的方法，並給定不同的字碼。

## 第七章 結論與未來展望

### 7.1 結論

由於每個人的發音皆不同，而嵌入式語音辨識系統在研究上又尚未成熟，若要增加辨識成功度就必須投入大量人力建立語音資料庫，加入訓練。但是在資源有限的嵌入式系統上是一個非常大的負擔，所以儲存空間必須謹慎使用。在嵌入式系統中擷取訊號資料並作處理時需要大量的計算，特別是當環境中有非常多的雜訊時，內部的濾波處理，但男女聲音的頻率又大不相同，每一個人說出菜單的時間也大不相同，就算是同一個人所說出的同一句話，第一次與第二次的特徵值也絕對不一樣，所以必須要蒐集大量的聲音以便提高辨識率。本系統使用預錄聲音的方式進行輸出，並且整合多單詞的測試，在室內的最高平均辨識率可達到 82.6%，其成功率相當高。本系統能夠為使用者提供一個操作簡單、快速、低耗能的環境，而我們將此系統應用於點餐服務機器人上，具有非常好的結果，同時它也能夠很容易的移植到其他機器人語音辨識的工作。

## 7.2 未來展望

我們希望在未來能夠在智慧型機器人上加入語音合成，取代較呆板的預錄方式，蒐集更多中文聲音加入語音資料庫，以較有變化的方式來表現音節的順序，以簡單的例子來表示：例如“目的”和“的確”，雖然表現出的文字是一樣的，但是聲音卻完全不同。除了能加入語音合成之外，希望還能夠加入麥克風陣列的研究，此研究除了可以幫助取出更清晰的信號之外，還能夠有導航的功能，可利用行走系統與避障功能，準確找出使用者的位置和方向。最後將系統升級為DSP(TMS320C6713)，除了提升運算速度同時也能加入更多的演算法，提升辨識率，讓此研究在機器人的應用與配合上能夠更具有智慧。

## 參考文獻

- [1] Pei Ding and Lei He, “Robust Mandarin Speech Recognition in Car Environments for Embedded Navigation System”, IEEE Transactions on Consumer Electronics, Vol. 54, pp. 584-590, 2008.
- [2] In-Chul Yoo and Dongsuk Yook, “Automatic Sound Recognition for the Hearing Impaired”, IEEE Transactions on Consumer Electronics, Vol. 54, pp. 2029-2036, 2008.
- [3] Hyeopwoo Lee and Sukmoon Chang, “A Voice Trigger System using Keyword and Speaker Recognition for Mobile Devices”, IEEE Transactions on Consumer Electronics, Vol. 55, pp. 2377-2384, 2009.
- [4] M. Paulik, S. Stuker, C. Fugen, T. Schultz, Thomas Schaaf, and A. Waibel, “Speech Translation Enhanced Automatic Speech Recognition”, in ASRU, San Juan, Puerto Rico, 2005.
- [5] G. Doddington, M. Przybocki, A. Martin, and D. Reynolds, “The NIST Speaker Recognition Evaluation – Overview, Methodology, Systems, Results, Perspective”, Speech Comm., Vol. 31, pp. 225-254, 2000.
- [6] H. Chung and I. Chung, “Memory Efficient and Fast Speech Recognition System for Low-Resource Mobile Devices”, IEEE Transactions on Consum. Electron., Vol. 52, No. 3, pp. 792-796, 2006.
- [7] H. VanHamme, “Robust Speech Recognition Using Cepstral Domain Missing Data Techniques and Noisy Masks”, in Proc. IEEE Int. Conf. Acoustic Speech Signal Process, pp. 213–216, 2004.

- [8] H.-Y. Cho and Y.-H. Oh, "On The use of Channel-Attentive MFCC for Robust Recognition of Partially Corrupted Speech", *IEEE Signal Process. Letters*, Vol. 11, No. 6, pp. 581–584, 2004.
- [9] J.-M. Valin, J. Rouat and F. Michaud, "Microphone Array Post-Filter for Separation of Simultaneous Non-Stationary Sources", in *Proc. IEEE Int. Conf. Acoustic Speech Signal Process*, pp. I–224, 2004.
- [10] Shiau-Chuan Wang, "Digital Processing of Speech Signals", CHWA Co. Ltd., 2008.
- [11] R. Singh, M. L. Seltzer, and B. Raj, "Speech in Noisy Environments: Robust Automatic Segmentation, Feature Extraction, and Hypothesis Combination", in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 273–276, Apr. 2001.
- [12] Waleed H. Abdulla, David Chow, and Gary Sin, "Cross-words Reference Template for DTW-based Speech Recognition Systems", *Conference on Convergent Technologies for Asia-Pacific Region*, Vol. 4, pp. 1576–1579, 2003.
- [13] Anup Kumar Paul, Dipankar Das and Md. Mustafa Kamal, "Bangla Speech Recognition System Using LPC and ANN", *Department of Information and Communication Engineering*, Vol. 4, pp. 171–174, 2009.
- [14] Chia-Feng Juang, Chyi-Tian Chiou and Chun-Lung Lai, "Hierarchical Singleton-Type Recurrent Neural Fuzzy Networks for Noisy Speech

Recognition”, IEEE Transactions on Neural Networks, Vol. 18, No. 3, pp. 833–843, 2007.

[15]Marco Kühne, Roberto Togneri and Sven Nordholm, “Robust Source Localization in Reverberant Environments Based on Weighted Fuzzy Clustering”, IEEE Signal Processing Letters, Vol. 16, No. 2, pp. 85–88, 2009.

[16]Mau-Rung Lin, “Digital to Signal-Procession-Form Speech to Digital Radio”, CHWA Co. Ltd., 2008.

[17]Yi-Jeng Meng, “Digital to Signal Processing”, FLAG Co. Ltd., 2008.

[18]陳明榮, “PC 電腦語音辨認實作”, 旗標出版社, 民國 83 年.

[19]張興政、周明鴻、許炳煌, 語音辨識系統, 逢甲大學自動控制工程學系, 中華民國, 台中市, 96 年 4 月。

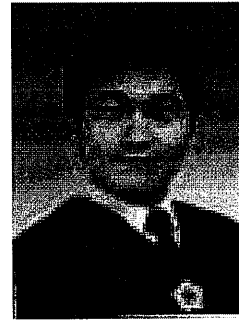
## 作者簡介

作者簡介：

姓名：朱銘傑 (Ming-Jie Ju)

出生日期：1987年01月25日

籍貫：臺灣省台北市



學歷：

2002/9~2005/6 臺北市立松山工農 電子科

2005/9~2009/6 私立中華科技大學 四技部 資訊工程系

2009/9~2011/7 國立勤益科技大學 電子工程研究所 資料組

研究經歷：

1. 自動節能割草機器人。
2. 第二代點餐服務型機器人。
3. 第三代點餐服務型機器人。
4. 第四屆凌陽盃小小鋼鐵人-佳作。



論文著作：

1. Guo-Shing Huang, Sheng-Chih Yang, Ming-Jie Ju and Shih-Hung Kao, “Lawn Mower Driving System Using Microcontroller-based Wheeled Mobile Robot”, The 5th Intelligent Living Technology Conference, pp.36-41, National Chin-Yi University of Technology, Taichung, Taiwan, June 4, 2010.
2. Guo-Shing Huang, Sheng-Chih Yang, Ming-Jie Ju, “Application of Meal Service Robot Using ASR”, The 4th Cross-Strait Technology, Humanity Education and Academy-Industry Cooperation Conference, pp. 電 36-1~9, National Chin-Yi University of Technology, Taichung, Taiwan, Dec. 2, 2010.

