中華民國八十四年全國計算機會議論文集
Proceedings of National Computer Symposium 1995

歌唱聲至樂器聲之即時轉換系統
A Real-Time System for Singing-Voice to Instrument-Music Conversion

古鴻炎　　譚百華
Hung-yan Gu and Martin Tarn

國立臺灣工業技術學院 電機系
Department of Electrical Engineering
National Taiwan Institute of Technology
Taipei, Taiwan, R.O.C
E-mail: root@guhy.ee.ntit.edu.tw

摘要

本論文提出一個把人的歌唱聲音轉換成樂器聲音的即時系統，它是由三個主要部份所組成，第一部份負責人聲基頻的求取，第二部份負責基頻到音符的轉換，第三部份則負責樂器聲音的合成。關於第一部份，我們提出了一個基頻求取的快速演算法，以及一個基頻平滑化的作法。關於第二部份，我們研究出一個不錯、可行的作法，以把一序列的基頻數值轉換成一串的音樂音符。關於第三部份，我們採用了 FM 合成技術去製作即時的樂器聲音合成軟體。我們的即時轉換系統，使用了 486-80 相容個人電腦與一片16 bits 之 AD/DA 介面卡，由實驗的結果顯示，從歌唱信號進入到樂器信號出來，延遲時間可縮減到 0.5 秒，並且輸出樂器信號的音高、音長、音強可相當忠實地追隨輸入的歌唱信號。

關鍵詞：歌唱聲，電腦音樂，信號處理，即時系統

Abstract

In this paper, a real-time system is designed and implemented to convert singing-voice to instrument-music. Within the system, three of the components are the most important. The first component estimates the lengths of the pitch periods in the singing-voice. The second performs the conversion from a sequence of pitch periods to a sequence of music notes. The third synthesizes the sound signal of a music instrument according to the notes. To estimate pitch periods' lengths, a new algorithm has been proposed. Also, a smoothing algorithm is proposed to eliminate occasional errors in pitch length estimation. To convert pitch length sequence into note sequence, a practical method suitable for real-time implementation has been developed. To synthesize the sound signal of an instrument, the technique of FM synthesis is adopted to write the software.

The real-time system's hardware includes two key components, i.e. ,an 486-80 compatible personal computer and a 16 bits AD/DA interface card. After practical test, it is found that the system's delay between the input singing-voice and the synthesized instrument-music is 0.50 second. Also, the frequency, duration, and intensity of the synthesized music signals follow the singing-voice very faithfully.

Keywords: singing voice, computer music, signal processing, real-time system

1. 前言

彈奏樂器需要的不僅是靈活的手腳，而且需要決心與毅力去作長時間的練習，很多人花費了許多的時間與精力才精通一項樂器，但是有更多的人無法完成其夢想(如肢體殘障者)，如果能以歌唱聲作爲輸入來產生樂器聲，應該可以幫助很多人完成其夢想。

要即時的將歌唱聲轉換成樂器聲，乍看之下會認爲只要將歌唱聲的基頻算出，並據以合成樂器聲就可以了，但事實不然，因爲音樂旋律是有一定的組成結構的，簡單說來旋律是由一序列的音符所構成，而一個音符具有音高，音強，音色，音長等特性[1,2]，因此我們需先將歌唱聲轉換成一序列的音符，然後再依據音符的資料去合成樂器聲。

要把即時輸入的將歌唱聲信號轉換成一序列的音符會碰到二個較嚴重的問題，即音符之音長的決定，與相鄰音符之邊界的決定(稱爲音符切割之問題)。關於音符音長的問題，因爲我們要做出

一個即時的系統，所以一個音符的音長不能等到一個音符完全結束時才計算出來，否則原始的旋律就會被破壞，以圖1爲例，原始歌唱信號爲一個

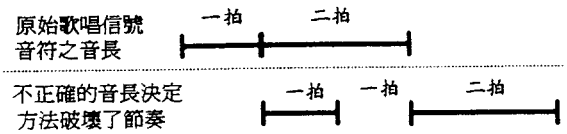


圖1 音符之啓始時間與音長

一拍的樂音後接著一個二拍的樂音，如果等一個音符結束才去決定音長，就會算出一個一拍的音符，接著一個一拍的休止符，再接著一個二拍的音符，而破壞了原始的旋律與節奏。關於音符切割的問題，因爲歌唱聲信號和語音信號一樣有共發聲(coarticulation)的現象，這使得相鄰音符間的邊界不易決定，並且邊界若不能正確決定(即音符不能正確切割)，則會破壞原始的旋律，例如 "妹妹背著洋娃娃"，其旋律是"So So Mi Re Mi Re Do",若不能將"妹妹"這個有共發聲現象的發音音高"So So"正確地切成兩個"So"的音符，而只切成一個兩倍時間長的單一音符"So"，則原始的旋律就被改變掉了。

我們嘗試運用電腦音樂與語音信號處理這二個領域的知識，來發展一個即時的歌唱信號到樂器信號的轉換系統，這是一個嶄新的嘗試，在進行研究的這一段時間尚未發現有類似的論文發表。我們建造的即時歌唱信號到樂音信號的轉換系統含有三個主要的處理方塊，如圖2 所示。

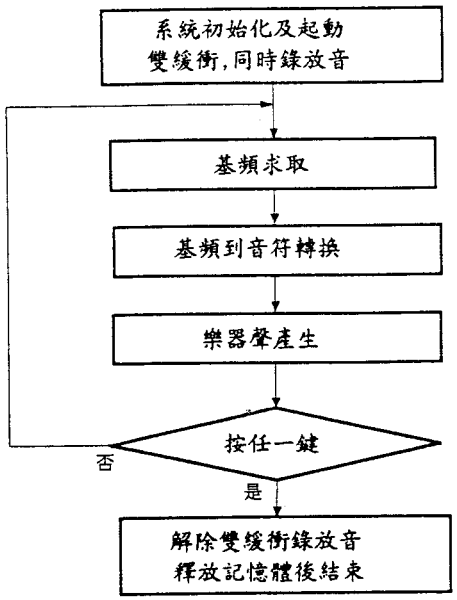


圖2 系統主流程圖

第一個主要方塊負責將歌唱信號轉換成一序列的基頻數值，實際上的作法是在時域上找尋信號的波峰位置以決定基週的時間長度。第二個主要方塊要依據一序列的基頻數值、能量值、及零交越率值去轉換出一串的音符，一個音符的相關資料包括啓始時刻、結束時刻，以及此音符之音高、音強數值。第三個主要方塊則要依據音符去合成樂器聲信號，我們採用了 FM 合成技術。

本文底下各節的內容是，第二節介紹歌唱信號的基週求取方法，與基週平滑方法。第三節介紹基頻值序列至音符序列的轉換方法。第四節說明樂聲的合成方法。第五節說明系統組成及實驗的情形。第六節是結語。

2. 基週求取

要將歌唱信號轉換成樂器聲信號的第一個處理步驟是，先將歌唱信號轉換成一序列的基頻數值，我們稱此步驟爲基週求取。歌唱信號經由A/D卡進入電腦後，會被儲存於緩衝區中，基週求取就是要持續地從緩衝區中拿出信號來分析，以求出基週值，求取基週的流程圖如圖3 所示，接著我們介紹流程圖裡各個方塊的功能。

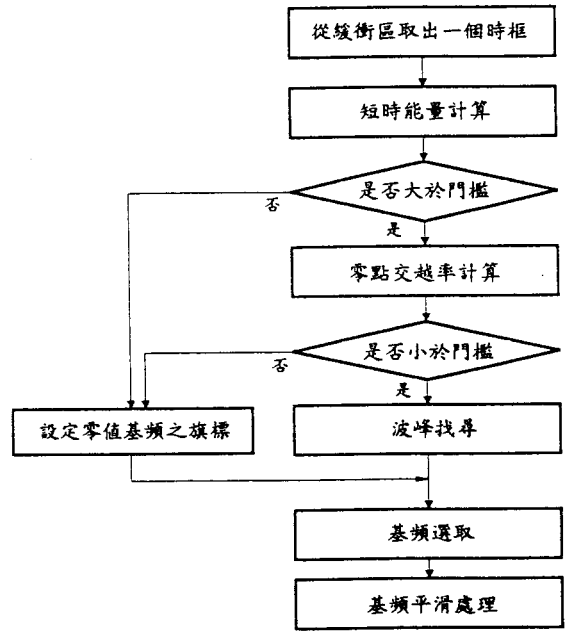


圖3 基週求取之流程圖

時框長度的選取需要考慮的因素爲歌唱聲最低基頻、即時反應需求、與取樣頻率(本研究設定取樣頻率爲 11,025Hz，且每個樣本爲16 bits)，考慮這些因素後我們就設定一個時框的長度爲500

點，並且時框是連續而不重疊的。爲了快速反應信號的變化，我們將一個時框分成兩個部份，分別去計算短時能量(定義其爲振幅絕對值之和)，只有在兩個部份的能量都大於門檻時，才認爲此時框中會有週期性信號，能量的門檻值的實際設定爲 100,000。接著，我們以零點交越率來進一步判定一個時框中的信號，是否爲週期性信號或是靜音與其它雜訊，所設定的門檻是 100次/每個時框。

直接在時域上作波峰之偵測，是爲了快速求得基週長度以符合即時處理的要求。過去王教授等人曾提出一種基週長度求取的方法[3]，但是，在這裡我們提出另一種方法來求取，其主要理念是以雙門檻做爲peak(波峰)點的偵測的依據，我們的演算法會先從整個時框中找尋最大值 Max 與最小值 Min ，當要選定一個peak點的位置時，除了第一個peak點就是最大值 Max 所在的位置外，其它的peak點在找尋的過程中都必需通過雙門檻的限制，我們才選取它爲一個peak點的位置。以圖4 的波形來說，設 A 點與 H 點是本時框的

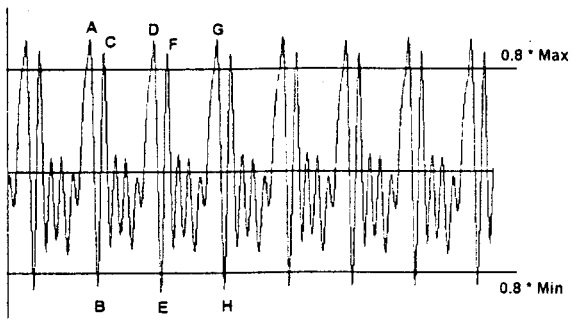


圖4 peak點偵測方法之說明圖

Max 與 Min 點，則演算法會先記錄 A 點的位置，並以 A點的位置當起點再往右邊尋找下一個 peak 點，但是我們限定要找到一個低於 0.8*Min 點(即B點)之後，高於 0.8*Max 的點(如C，D點)才會被記載(其振幅與時間位置)下來，並且振幅大的會把振幅小取代掉，如D點會取代C點，接著當再次碰到一個小於 0.8*Min 的點(即E點)時，我們就將B點E與點之間具有最大振幅且超過 0.8* Max 的點(即D點)設定爲下一個找出的peak點，然後將 Max 值更新爲D點的振幅，Min 值更新爲E點振幅，如此繼續下去。此演算法要從一個時框偵測出 5個 peak 點，最多只需對整個時框做兩次掃描，第一次是找最大值 Max 與最小值 Min ，第二次是實際去尋找。由於我們的程式先前在短時能量與零點交越率計算時已經對整個時框作過一次掃描，所以在此就可以直接作peak點尋找的動作。

一個時框經過上面的peak點找尋過程後，接著需將peak點的位置由小到大作一次排序，再由相鄰兩個peak點的間距去除取樣頻率而求得基頻值。由於我們規定在一個時框中要取得四個基頻的數值，因此對於一些無法求出四個基頻的時框，必需做一些額外的處理，我們以兩種情況來說明：(1)第一種情況是，求出的基頻數值少於4個，其發生原因是時框中的信號基頻太低,此時就以先前求出的最後一個基頻值來填補。(2)第二種情況是，這個時框沒有經過前面的peak的尋找過程，其原因是這個時框的零點交越率太大或能量太小，此時就直接設定此時框的四個基頻值都爲零。

我們在實驗時發現，如果將前面程序所產生的基頻值直接拿來使用，則所合成的樂器聲音常會有跳動的現象，造成此現象的因素包含如下二點：(1)歌唱者本身的歌唱信號的基頻並不穩定(抖音)，(2) peak點偵測演算法在處理較複雜的信號波形時產生了誤判。爲了改善前述的情況，我們提出了一個基頻平滑處理的方法，此方法的主要觀念是，對每個基頻值，都把它和前面的八個基頻放入一個9元素之陣列，先做一次中值濾波，再作一次均值濾波，在中值與均值濾波的過程中，都先將零頻值(表示靜音或雜訊)排除，以免造成錯誤影響。

在說明此方法前先定義四個陣列：A 陣列是一個能夠存放本次基頻值及前面八個基頻值的陣列；B 陣列是一個暫存陣列，其功用是在不破壞 A 陣列元素的順序下，能將 A 陣的元素加以排序；C 陣列存放經過中值濾波的九個基頻值；D 陣列存放經過平滑處理後的四個基頻值。平滑處理的較詳細步驟如下：

<1> 以步驟2至11處理一個時框中求出的四個基頻值。
<2> 將 A，C 與 D 三個陣列中的內容向左移一位，以將每個陣列中的第一個元素淘汰而空出陣列的最後一個位置。
<3> 將本次時框中的一個基頻值置入A陣列的最後一個位置。
<4> 如果 A 陣列中的第五個元素是零且第四個與第六個元素有一個爲零，則將 C，D 陣列的最後一個元素設爲零，否則執行步驟 5 至 11。
<5> 將 A 陣列的內容複製一份到 B 陣列。
<6> 從 B 陣列的前面開始找尋第一個連續不爲零的區間。
<7> 在 B 陣列中連續不爲零的區間內求取中值 Mid 。
<8> 將 Mid 值放入 C 陣列的最後一個元素 。

<9> 從 C 陣列的最後一個元素往前找尋一塊連續不爲零的區間。

<10>在 C 陣列中連續不爲零的區間求取均值 Ave。

<11>將 Ave 放入 D 陣列的最後一個元素。

採用前面的平滑處理程序，會使得系統對本次時框的反應時間延後，這是因爲平滑處理的基準點是位於9個元素之陣列的第五個元素，這個元素與本次求得的基頻值差了四個位置，也就是說差了一個時框。雖然平滑處理會造成延遲，但可以使得基頻跳動的情況獲得改善。

3. 基頻至音符轉換

歌曲之旋律可看成是由一序列的音符所串成，因此，下一步就是要從基頻值序列去求得對應的音符序列，這樣的問題我們稱爲音符之確認與切割。本研究因爲要達到即時轉換的目標，所以定義一個音符的相關資料爲音符起始時間、音符結束時間、音高、音強與音色，即把音長改成起始與結束的時間，而一旦偵測到音符起始了，就去合成該音符的樂器聲。

要決定一個音符何時起始與結束，我們所依據的線索是歌唱信號的短時能量、零點交越率〔9,10〕、及基頻值的變化，因爲在第二節基頻求取階段已經依據能量與零點交越率把靜音和雜訊時框的基頻值設爲零，所以在本階段，我們可直接去看上一階段輸出的基頻值序列來決定一個音符的起始與結束時間，當碰到基頻值爲零時，就知道是一個音符的結束了，或尚未有音符起動。此外，我們還依據基頻值的變化來作判斷，其作法是將音符的起始與音符的結束都設立基頻值變化的門檻，在音符起始點方面，爲了正確地反應音符的音高，只有當連續的基頻值變化都在一個較小的範圍內，才認爲是一個音符的起始時刻；而在音符的結束點方面，爲了減少不必要的跳動，只有在基頻值變化連續超過一個較大的門檻值時，我們才認爲是一個音符的結束時刻。

應用基頻值的變化來切割出音符，我們就可解決歌唱信號裡，單一音節對應多個音符的情況，以及音節與音節間有共發聲而波形相連的情況。整體來說，我們提出的音符確認與切割之演算法可由圖5 的流程圖來說明，在圖5 中，有幾個區塊需要再加以詳細說明，現在分述如下:

取對數並轉成音階:

若基頻值 y 爲0，則設定音階值 s 爲0；否則令音階值 s 爲

$$s = \log(y) / (\log(2.0)/12) \qquad (1)$$

此公式用來把線性頻率尺度轉換成十二平均律音階的尺度。

一個音階起動了:

此區塊是要依據一個狀態變數(音符行進旗標)來作判斷，這個狀態數值的初值爲零，表示未起動，它可由其它區塊去設定或解除。

輸出音高、音強資料:

音高: 將音階值 s 轉換成頻率值 f

$$f = \exp(s*\log(2.0)/12) * 4 \qquad (2)$$

音強: 以時框中的最大振幅 Max 爲參考音強；音色: 以使用者指定的樂器爲音色。

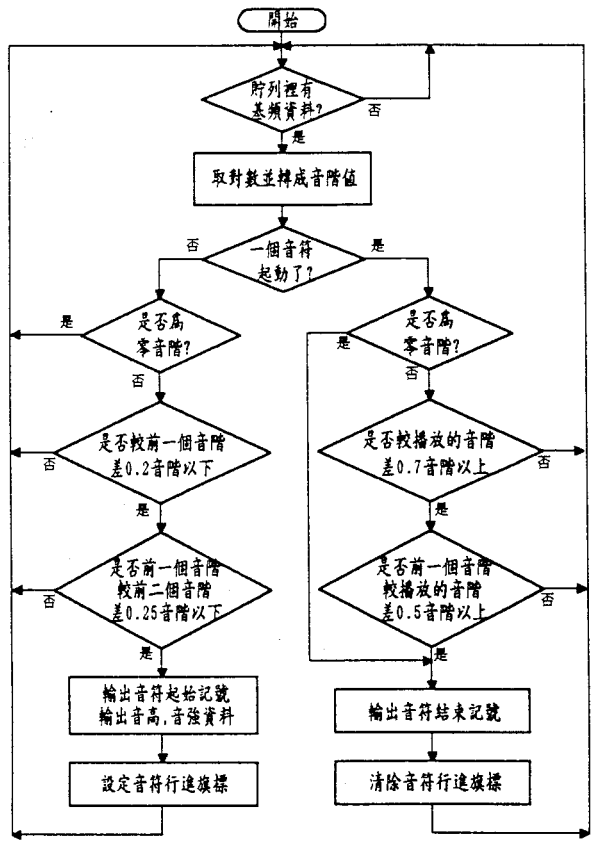


圖5 音符確認與切割之流程圖

4. 樂音產生

FM 技術是常被使用的電子音樂合成技術，從電子合成之鍵盤樂器到多媒電腦的聲訊配備(聲霸卡)，都運用此種技術來產生樂音信號。其基本原理就是一個載波接受一個調變波的調變，詳細的公式〔4,5〕如下:

$$X(n) = A*I(n)*\sin\{2\pi f_c nT + I(n)*\sin(2\pi f_m nT)\} \quad (3)$$

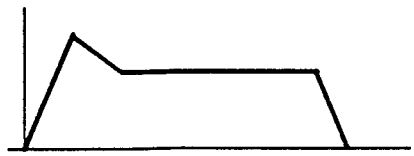
其中X(n)表示由 FM 技術產生的樂器聲波形，A是控制振幅大小之常數，I(n)是一個樂音從觸發狀態經過平穩狀態到最後結束狀態的包絡函數(envelop)，fc是載波的頻率，fm是調變波的頻率，n 是取樣點，T 是取樣週期。

此種技術的優點在於：(1)結構簡單，容易實行，並且可相當傳眞地模倣出某些自然樂器的音色；(2)可藉由不同的參數組合，創造出非現實樂器的樂音。而其缺點則是：(1)對於一個特定的樂音音色，尙未有一個固定的程序，可用以決定 FM 合成公式裡所需的各項參數的數值；(2)合成的音色與眞實樂器的音色還是有一點差別。

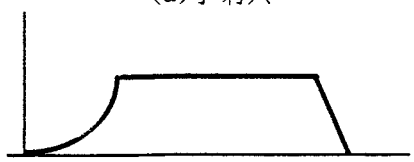
當使用 FM 技術合成一種樂器的樂音信號時，我們需設定 FM 合成公式裡的參數數值，這些參數會影響所要產生樂音的四個特性，即音高、音強、音色與音長。

要設定所合成樂音的音高，我們需依據基頻的數值，去決定fc和fm的數值，但還需配合音色設定裡，基頻與載波頻率比例和基頻與調變波頻率比例的關係。影響音強的參數只有公式(3)裡的 A 參數，藉由改變 A 的值來改變整個波形的振幅。

影響音色的參數爲：(1)包絡函數 I(n)，(2)基頻對載波頻率fc的比率，(3)基頻對調變波頻率fm的比率，當上面三個項目都確定時，就可確定所要合成之樂音的音色。例如要合成小喇叭的音色時，我們需令I(n)函數具有如圖6(a)[6]所示的形狀，基頻與載波頻率的比例設爲1:1，基頻與調變波頻率的比例也要設爲1:1；而要合成黑管的音色時，我們需令I(n)函數具有如圖6(b)[6]所示的形狀，基頻與載波頻率的比例要設爲1:3，基頻與調變波頻率的比例要設爲1:2。



(a)小喇叭



(b)黑管

圖6 包絡函數圖

音長就是公式(3)中，取樣點 n 的範圍，當產生的取樣點愈多，則音長愈長。與取樣點 n 有關參數還有I(n)函數，前面曾提到I(n)函數形狀與音色有關，但現在我們也要考慮如何在保持原有音色的條件下延長或縮短I(n)函數的時間範圍，以決定合成樂音的音長，由觀察圖6(a)與6(b)可知，小喇叭與黑管的I(n)函數在時間軸的中間部分呈現穩定的狀態，因此，我們便以延長或縮短中間的部分的方式來調整音長，而實際的實驗也顯示這樣的作法是可行的，詳細來說，我們設定小喇叭的I(n)函數的公式是：

$$I(n) = \quad (4)$$

$n/1102$, if $0 \le n \le 1102$ (起始狀態1)

$1-(0.25*(n-1102)/1102)$, if $1102 \le n < 2204$ (起始狀態2)

0.75, if $2204 \le n$ (穩定狀態)

$0.75*(1-(n/500))$, n重設爲0, $0 \le n \le 500$ (結束狀態)

黑管的I(n)函數的公式是：

$$I(n) = \quad (5)$$

$(n/1102)^2$, if $0 \le n \le 1102$ (起始狀態)

1, if $1102 \le n$ (穩定狀態)

$0.75*(1-(n/500))$, n重設爲0, $0 \le n \le 500$ (結束狀態)

5. 測試實驗

本系統的硬結構如圖7 所示，是以一部 AMD 486-80 個人電腦爲控制中心，歌唱信號輸入麥克風後，經由放大器傳給插在電腦內的 A/D 介面卡，而轉換成爲數位信號交給電腦處理，我們採用 Borland c++ 3.1[7,8]來發展信號處理的軟體。在相反方向，依照公式(3)(4)(5)得到的數位樂器聲信號，就交由介面卡作 D/A 轉換成類比信號，再經由放大器與揚聲器發出樂音。以上的程序，從歌唱信號的進入，到實際樂音的產生，都是在很短的時間內完成，最小延遲時間已減少到 0.50 秒，而且可以連續無間斷地處理進入的歌唱信號。

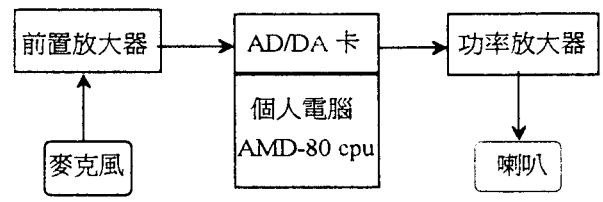


圖7 系統的硬體架構

爲了測試系統的動作是否正確，我們暫時將即時轉換系統的信號輸入與輸出來源，改成爲檔案輸入與輸出。

音符切割測試

圖8 是原歌唱聲"妹妹背著洋娃娃"之信號檔的波形圖，圖9 是"妹妹背著洋娃娃"之信號未經過基頻平滑處理的樂器信號合成的波形圖，圖10 則是經過基頻平滑處後的樂器信號合成的波形圖。由圖9 可以很清處的發現，未經平滑處理會由輸入的信號切得12個音符，而由圖10 可以發現經過平滑處理則只切得七個音符，從這個測試實驗可發現，經過平滑處理會減少音符誤判而產生的跳動現象。此外，我們也可以由歌唱聲"妹妹背著洋娃娃"的信號波形裡發現，"妹妹"與"著洋娃娃"都產生了共發聲現象，而我們的音符切割演算法，在此種共發聲現象中，仍能正確的將音符切割出來。

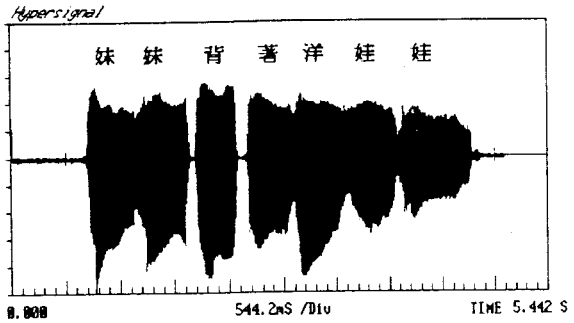


圖8 "妹妹背著洋娃娃"原歌唱信號的波形

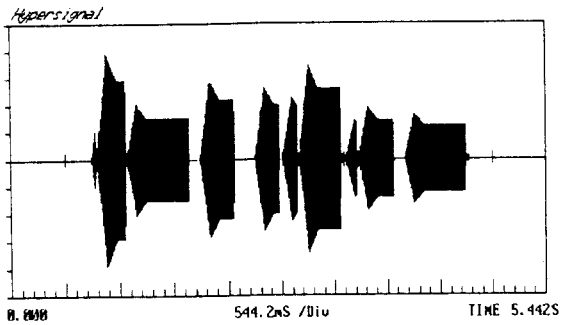


圖9 未經過基頻平滑處理的合成樂器聲波形

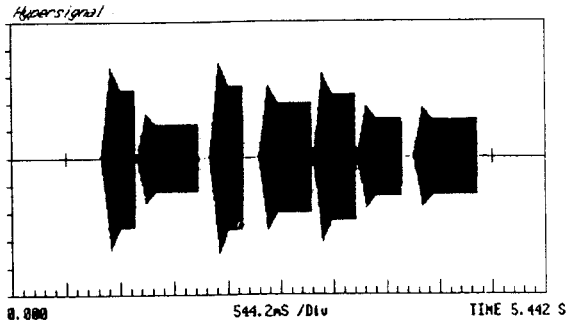


圖10 經過基頻平滑處理的合成樂器聲波形

音高測試

表1 是 "妹妹背著洋娃娃" 歌唱信號的音高，未經過平滑處理與經過平滑處理的比較。由於實際產生的樂器聲的音高是將所算出的基頻值乘上四倍，以符合一般樂器演奏的音高，所以表1裡的音高是眞實歌唱聲音高的四倍，至於表1 的歌唱聲音高，則是經 FFT 頻譜分析後，目測頻譜圖上的基頻值而得到的。由表1 可知，經基頻平滑處理求得的音高值，比未經平滑處理的具有較小的誤差。

表1 音高比較表

	妹	妹	背	著	洋	娃	娃
音符	So	So	Mi	Ra	Mi	Ra	Do
歌唱聲音高(Hz)	138	146	119	113	123	108	98
未平滑處理 (Hz)	528 564	604	472	436 456	504	456 412 436	388 420
已平滑處理 (Hz)	559	595	485	452	487	445	393

音強測試

在音強方面，我們是在偵測到一個音符起動時，就以當時的信號振幅來設定合成的樂器聲的強弱，這種作法雖然簡單，但實際上的效果不錯，這可由比較圖8 與圖10 而得知，合成的樂器聲信號的強弱都能跟隨歌唱聲信號的振幅成比例地變動。

6. 結語

要即時的將歌唱聲轉換成優美的樂器樂音，可以說是一個不簡單的問題，牽涉到演算法的發展及上機實做，但是，能將科學的知識應用在藝術的課題，實在令人雀躍。研究的過程，我們經過了一序列的試聽與改進的階段，雖然辛苦，但最後總算獲得了一些成果。

本論文主要的貢獻在於，提出了一套完整而且明確的方法，並實作出一個即時的轉換系統，以將歌唱聲信號轉換成樂器聲信號。我們將此轉換問題分成三個主要的子問題：即基頻求取、基頻至音符轉換、與樂器聲的產生。

基頻求取：我們提出了一個在時域上快速找尋波峰點的演算法，以及一個對有錯誤存在之基頻值序列作平滑處理的方法。

基頻至音符轉換：在即時處理的要求下，我們定義一個音符的相關資料爲音高、音強、音色、音符起始時刻與音符結束時刻等，並且對這五個資料的求取，提出可行的作法。關

於音符之切割，我們提出以能量、零交越率、與基頻變化來作判斷的方法。

樂器聲之產生：我們採用前人發展的 FM 樂音合成技術，但是加以修正並寫成軟體，以符合即時系統的需求。

雖然，我們已經發展出一種可行的作法，但是仍有一些地方可再改進，包括：(1)如何減少時間的延遲，(2)如何將現有的一個音符一個固定的音強，改變成一個音符的音強可以漸變的(由弱漸強)，(3)如何將現有的獨奏(單音)樂器聲，加上節奏、合弦與其它樂器聲成爲一個合奏樂器聲。

參考文獻

[1] 謝寧，音樂的科學原理，68年5月。

[2] 林肇華、林肇富釋，音樂欣賞教程，74年8月。

[3] J. F. Wang,, et al., "A Hierarchical Nerual Network Model Based on a C/V Segmentation Algorithm Isolated Mandarin Speech Recognition", IEEE Trans. Signal Processing, Vol. 39, No. 9, pp. 2141-2146, Sept. 1991.

[4] J. A. Moore, "Signal Processing Aspects of Computer Music: A Survey", Proceedings of the IEEE, Vol. 65, No. 8, pp. 1108-1137, Aug. 1977.

[5] F. R. Moore, Elements of Computer Music, Prentice-Hall, 1990.

[6] J. M. Chowning, "The Synthesis of Complex Audio Spectraq by Means of Frequency Modulation", Journal of the Audio Engineering Society, Vol. 21, No. 7, pp. 526-534, 1973.

[7] Borland, Borland C++ Programmer's Guide, 1990.

[8] Borland, Turbo Debuger 3.0 User's Guide, 1988.

[9] 陳榮貴，應用於國語語音辨認之自動切割技術之研究，國立台灣大學電機研究所碩士論文，1988年6月。

[10] 洪鴻文，國語連續語音切音技術之研究，國立台灣工業技術學院電子系，碩士論文，1990年7月。