

智慧型自動吉他伴奏系統之實現

趙春榮

黃建生

南台科技大學 電機系

tang@mail.stut.edu.tw

M9520221@mail.stut.edu.tw

摘要

本論文目的在實現一個智慧型自動吉他伴奏系統，希望能應用我們所設計的自動吉他系統，結合 MIDI 技術以及智慧型的學習，擴充其具有伴奏的功能。基於實用上的考量，我們將提供兩種模式的伴奏效果。首先第一種模式，主要為演奏者配合伴奏系統模式，演奏者可以透過 MIDI 樂器演奏樂曲，而伴奏音樂的每一個音符的發音時間，都可以開放由演奏者來修改或設定。在這種模式下，雖然演奏者可以決定伴奏樂曲的每一個音符的節奏，可是伴奏樂曲每次伴奏時，是不可以即時作改變的，是以此模式又稱為演奏者配合伴奏系統模式。第二種模式的目的則是希望伴奏系統能配合演奏者，主要分為學習 (Learning) 和回想 (Recalling) 兩過程。在學習過程中，此時的伴奏系統將自動撥放，而系統將自動分析演奏者所彈奏的主奏內容，也就是說當演奏者很隨興的每次慢慢加快或放慢演奏速度時，伴奏系統都能夠有智慧地學習。而在回想過程中，伴奏系統將根據之前學習過程的學習經驗，有能力跟隨演奏者的節奏，並預測出下一個伴奏音符的撥放時間，如此才能符合真正伴奏的意涵。為達成以上的目的，我們設計了一個有效率的類神經網路來學習樂曲中每一音節的演奏特性，以增進智慧學習的效能。藉由增加類神經網路的一些輸入資訊，我們得到了令人滿意的結果。我們深深的希望藉由此類音樂機器人計畫之實現，除了能增進電子、機械等方面之設計技術與人才培養外，亦能促進科技與藝術的結合。

關鍵詞：自動吉他、類神經網路、樂器數位介面、自動伴奏系統、科技與藝術。

1. 前言

在過去的國科會計畫支持下，我們已完成了一個自動吉他演奏系統[6]以及一個實用的自動吉他樂曲輸入系統[7]。如何賦與自動吉他新的應用與生命，則更是值得探討的問題。

對於演奏樂器的愛好者而言，單人獨奏固然有成就感，但是如果樂曲是二人或多人合奏，那麼演奏起來的效果，更是令人難忘的。藉由與他人合奏，可以切磋演奏技巧，增進節奏的敏感度，可以讓自己的演奏技巧突飛猛進。可是對某些人，或

某些時機而言，要能找到合適的愛樂同好進行合奏，並非易事。因此，所謂的「伴奏唱片」應運而生，例如“鋼琴協奏曲減一”(Piano-Concerto-Minus-One)，它能將主奏樂器——鋼琴的聲音去除，以方便彈奏者練習使用，並增進練習的情境與效果。然而，這種「伴奏唱片」所提供的練習樂曲，實在太少，而且伴奏的節奏是很固定不變的，對於廣大的初學者來說，並不適用。

若找不到以上的「伴奏唱片」，退而求其次，電子節奏器 (Tempo Controller) 也是一般愛樂者的選擇，基本上它能提供多種樂器音色、流行舞步節奏如：探戈、倫巴、森巴、迪斯可、搖滾等，以及任意的節奏設定。必須強調的是，電子節奏器還是主要以產生節拍音樂，甚至和絃音樂為主，無法表達任意的伴奏曲目。

以上無論是「伴奏唱片」或是「電子節奏器」，基本而言，由於伴奏音樂都固定，或是事先設定節拍速度，所以都可視為演奏者必須配合伴奏音樂的進行。這樣對於一般的情形而言，勉強可以接受，但是對於樂器初學者，或是樂器的精通者而言，一個能配合他們演奏的伴奏者，才是他們所希求的。本論文目的在實現一個「智慧型自動吉他伴奏系統」，目的就是利用我們過去所完成的自動吉他系統，克服以上缺點，來實現一個更具有智慧與彈性的自動吉他伴奏系統。

本系統除了應用過去發展 Midi 技術的成果外，也在過去的自動吉他樂曲輸入系統中，加入了伴奏功能，更參考了蘇文鈺教授所指導的碩士論文[4]中所提到的類神經網路的架構加以改良，結果顯示了本系統的智慧型學習能力確實能改善許多。

2. 系統介紹

圖1為本系統之架構圖，我們使用一台 YAMAHA E403 的 Midi 標準樂器，透過 USB 和電腦主機作連結。電腦主機另一端，則透過印表機埠外接 8255 卡，控制外部自動吉他的壓弦、撥弦、以及把位移動等。

首先介紹圖2的自動吉他硬體部分，主要包含機構的設計以及控制電路的設計。由於此系統採用氣壓作為動力來源，故實際操作時，尚需連接氣體壓縮機。

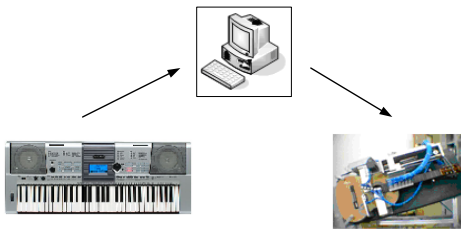


圖 1. 系統架構圖

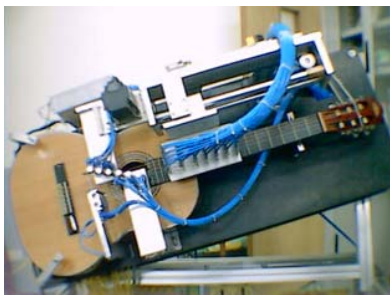


圖 2. 自動吉他

圖 3 為右手的撥弦機構，我們以六個氣壓缸配合特殊設計的撥弦結構，使氣壓缸的上下運動，達成撥弦的效果。

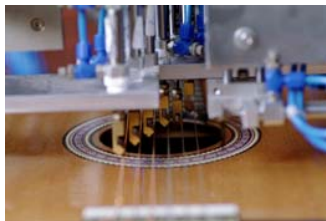


圖.3 自動吉他撥弦機構

而在左手壓弦機構的設計方面，由於實際演奏吉他時，左手擔任把位變換及壓弦的工作，而在同一時間，左手最多只能橫跨五個把位。我們同樣是利用氣壓缸的上下運動以達成壓弦的效果，如此說來，必須使用五排（每排六個，控制六條弦）的氣壓缸設計。然而由於吉他本身的把位寬度是隨著把位增加而減小的，為了使每個把位的每個點都能完成壓弦的動作，勢必需要再增加幾排氣壓缸才行。圖 4 為左手的壓弦機構設計，最後是以 7×6 的氣壓缸陣列，使得每一把位的每一壓弦點，都能順利的將弦壓下。此外，為減少壓弦時所可能產生的擊弦噪音，也在壓弦機構中，將會與弦接觸的部分，採用較為軟性而非剛性的材質。



圖.4 自動吉他壓弦機構

介紹自動吉他系統後，以下分別針對 MIDI 技術應用以及伴奏系統設計與學習等主題，分節敘述之。

2.1 MIDI 技術應用

MIDI 是 Musical Instrument Digital Interface 的縮寫，意思是「樂器數位介面」，它讓所有具有 MIDI 介面的裝置都能夠彼此連接。本論文運用了一些 MIDI 技術，以下分別說明之。

2.1.1 MIDI 多通道合音模擬

對於軟體工程人員來說，都應該有簡單的讓電腦發聲的程式經驗。以軟體為例，若要使電腦發出不同音階的聲音，可以利用作業系統所提供的 beep 函數，設定 beep 函數的發音頻率，以及持續的時間。可是這樣的電腦發聲，只能限於「單音」，且效果並不理想，必須要能展現「合音」的效果，才符合我們的理想，但這對一般程式軟體而言，並不容易達成。

數位樂器界面 MIDI (Musical Instrument Digital Interface) 自 1983 年誕生，它所製造出的電子音樂，的確給傳統音樂產生極大的影響。本論文也應用了 MIDI 所提供的音效模擬。由於古典吉他有六條弦，對 MIDI 而言，我們只需採用六個頻道 (Channel)，設定不同頻道發出不同的音階，就可以很輕易的產生合音的效果。

雖然本論文的目的是在於使用實體自動吉他作為伴奏樂器，不過由於 MIDI 技術能模擬數十種樂器音效，故加入 MIDI 音效合音技術，除了方便我們作系統模擬以外，也提供使用者另一種選擇，更能增加系統的實用性與娛樂性。MIDI 的音階 (Keynote) 以鋼琴為例，如圖 5 所示。鋼琴上的每一鍵都給定不同的 Keynote 值，我們對於吉他六條弦上每一把位的音，都找出其對應的 Keynote 值，即可進一步利用 VB 程式技巧，實現 MIDI 多通道合音模擬。



圖 5. 鋼琴鍵 Keynote 值圖示

本系統提供了如圖 6 的六線譜編輯介面，讓使用者輸入伴奏樂譜，在此介面下即可進行合音音效模擬。

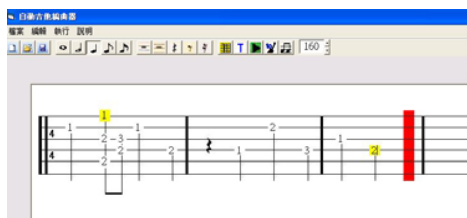


圖 6. 六線譜樂曲編輯介面

2.1.2 MIDI 檔案轉換

本系統亦提供對於伴奏樂曲轉換成 MIDI 檔案的功能。此功能的優點，在於使用者若對於系統所提供的伴奏樂音滿意，則可以將它儲存成 MIDI 檔案，方便日後隨時播放。MIDI 檔案結構是由 Chunk 所組成，包含了二種型式的 Chunk，分別為 Header Chunk 及 Track Chunk。一個 MIDI 檔只有一個 Header Chunk，用來記錄基本格式資料，其後可以接一個或數個 Track Chunk 用來記錄音符資料及 MIDI 命令資料。MIDI 檔案由於所佔空間小，且能模擬多種實際樂器的音效，所以廣受歡迎。

特別值得一提的是 MIDI 檔頭中，「分割數」與節拍時間的關係必須辨明清楚。例如若以四分音符為一拍，而分割數為 100，意思就是說一拍之內，再分割 100 個基本時間單位，這時間單位其實就是系統計時的中斷時間了。例如四分音符為一拍，若拍速為每分鐘 120 拍，則一拍應有 0.5 秒，這一拍再分割 100 份，則每一基本時間單位為 0.005 秒。而 MIDI 檔案中的音符間隔時間，就都是用分割數表示了，亦即如下式所示。

$$\text{音符間隔時間} = \text{分割數} * \text{基本時間單位} \quad (1)$$

在說明 MIDI 檔案轉換之前，必須先說明在自動吉他硬體控制方面，我們訂定了如表 1 的自動吉他硬體控制碼，以控制自動吉他的壓弦、撥弦、以及把位移動等動作。

表 1. 自動吉他硬體控制碼

名稱	功能
INTRO	起始碼，亦可輸入有關樂曲的文字介紹
Babcd	基本節拍時間= (a*1000+b*100+c*10+d) ms
Pab	移到第 (a*10+b) 把位 (左手)
Tab	(右手)撥弦後延後時間=(a*10+b)*基本節拍時間

Gad	休止時間=(a*10+b)*基本節拍時間
Lab	(左手)壓第 b 把位(相對目前把位)，第 a 條弦
lab	(左手)放開第 b 把位(相對目前的把位)，第 a 條弦
Rab	(右手撥)第 a 條弦，或同時撥第 a, b ... 條弦
D	Debug 碼，輸出一些訊息，供偵錯用
E	結束碼

利用以上的硬體控制碼，我們就能很方便的為每首吉他曲，寫出硬體控制碼樂曲檔案。以一首初學者所常彈的「C 大調小行板」[5,8]為例，以下為樂曲控制碼之部分摘錄。

INTRO B0170T01P01

L3251 R35R4R35R4 I3251R35R4R35R4
 L3353R35R4R35R4 I3353L3251R35R4R35R4

..... (略)

T01 I63 L3251 R35R4R35R4 I3251R35R4R35R4
 L3353R35R4R35R4 I3353L3251R35R4R35R4

I3251L23R26R4R26R4 I23L61R36R4R36R4
 I61L13R15R4R15R4 T04 I13L233251R2345 I233251

E

本系統另外提供一種樂曲輸入介面，較圖 6 的六線譜介面低階，可以讓使用者直接輸入硬體控制碼演奏樂曲，圖 7 所示為輸入「愛的羅曼史」硬體控制碼的範例。



圖 7. 硬體控制碼演奏介面

在圖 6 的六線譜編輯圖形介面下，亦可將伴奏曲目轉換成 MIDI 檔案。但實際上，無論是上一小節的「MIDI 合音音效模擬」，或是本小節的「MIDI 檔案轉換」，在完成這些動作之前，我們都是先將伴奏樂曲轉為硬體控制碼來表示的。由於我們在上小節 MIDI 音效模擬中，已經能夠取得所發的音（不論單音或合音）的音階，以及發音的時間長度，如此一來，再配合我們對 MIDI 檔案的了解，即可順利完成硬體控制碼轉 MIDI 檔案的工作。

2.1.3 MIDI 訊息擷取

MIDI 訊息傳輸是「串列」「非同步」傳輸的方式，速率為每秒 31,250 bits；又每個 MIDI 訊息（一組 MIDI 資料的單位）需要 10 bits 來記錄，所以 MIDI 每秒鐘可以傳送 3,125 個資料。由於 MIDI 所傳送的是數位控制訊號，並不是聲音訊號，故可以很有效率的表示音聲[2,9]。

以鍵盤樂器連接電腦為例，當鍵入 Do 時，會傳送 903C40 的碼，這 3 個 Byte 分別表示：

- 90: 發音碼 (Note On)
- 3C: 表 DO 音 (Note Number)
- 40: 按鍵時的力道 (Velocity)

若要改變下一個音為吉他的音色，此時便傳送 C019 訊息。

簡單而言，MIDI 訊息分為兩種格式，第一個格式為表示「命令」(Command) 的「狀態位元組 (Status Byte)」，第二個格式為配合命令的「資料位元組 (Data Bytes)」。MIDI 訊息通常以一個狀態位元組帶著二個資料位元組，共三個 bytes 組成一指令。若以訊息功能分類，則如圖 8 所示。

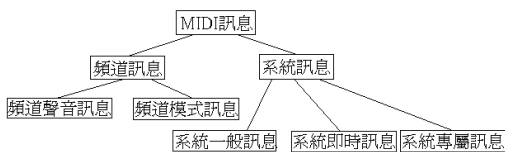


圖 8. MIDI 訊息分類

本論文中，我們使用 Mabry[13]所提供的 Midi 硬體連結程式，取得和電子琴溝通的方式，再自行撰寫 VB 程式，處理 MIDI 樂器傳來的主奏音樂訊息。如此即可了解演奏者所演奏的音符，再經系統的分析學習處理程式，決定伴奏的音符與時間。一開始使用電子琴彈奏音樂時，必須先指定 MIDI 輸入裝置 (MIDI In Device)，如圖 9 所示，再把圖 10 為軟硬體連結程式的初始畫面 Display Midi Input 的選項打勾，此時一旦開始彈奏電子琴時，此軟體將會把電子琴傳送進來的訊息以 16 進碼的方式顯示在初始畫面上，所以此軟體可以顯示出電子琴的 MIDI 十六進制碼。



圖 9. 指定 MIDI 輸入裝置

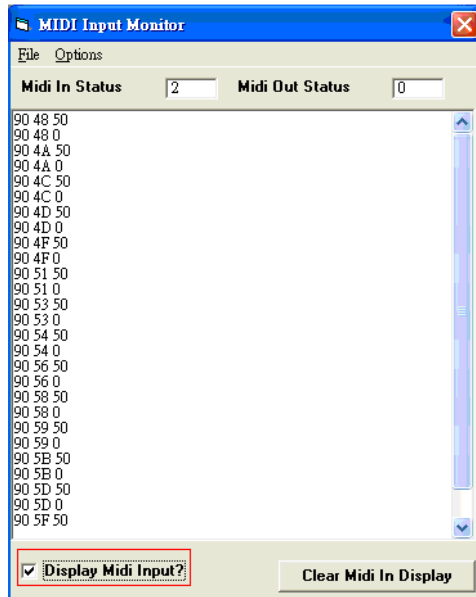


圖 10. 軟硬體連結程式的初始畫面

2.2 伴奏系統設計與學習

一個包含主奏與伴奏的樂譜，如圖 11 所示。本節將說明如何設計一個理想的伴奏系統，有關伴奏模式、音符群、學習樣本、類神經網路學習等主題，分節說明如下。



圖 11. 鋼琴樂譜範例 (曲名：靜夜星空)

2.2.1 伴奏系統

為了增加伴奏系統的實用性，我們提供了以下兩種模式。

模式 1：演奏者配合伴奏系統模式。

演奏者可以透過 MIDI 樂器演奏音樂；此外，伴奏音樂的每一個音符的發音時間，都可以開放由

演奏者來修改或設定。在這種模式下，雖然演奏者可以決定伴奏樂曲的每一音符的節奏，可是伴奏樂曲每次伴奏時，是不可以即時作改變的，所以我們稱為演奏者配合伴奏系統模式。

此種模式，主要是因應演奏者可能希望整個伴奏系統能緩慢或快速的演奏，或是演奏者對於某些音節不熟悉，希望這些音節的音符能延遲地出現。為了實現此伴奏模式，我們在過去所完成的圖 6--六線譜樂曲編輯介面中，進一步提供使用者設定伴奏樂曲每一拍的時間（預設值為 170ms），或設定某些伴奏音符的延遲時間。如此一來，樂曲的撥放，不再是固定的節奏，而是可以依據演奏者的喜好，事先設定的。

模式 2：伴奏系統配合演奏者（含「學習」、「回想」過程）

在這種系統下，伴奏系統有能力跟隨演奏者的節奏，也就是說當演奏者很隨興的每次慢慢加快或放慢演奏速度時，伴奏系統都能夠有智慧地，亦即有學習功能地，且即時地改變伴奏的節奏，如此才能符合真正伴奏的意涵。

本模式的適用對象，是進階的演奏者，故希望伴奏系統能配合他。本模式的實現，較前一種模式困難許多，以下說明均是針對此一模式。

2.2.2 音符群、間隔時間

由於主奏和伴奏經常會在同一個時間演奏多個音符，所以我們把主奏和伴奏同一個時間所撥放的音符視為一個音符群，圖 12 為音符群的圖例說明。



圖 12. 音符群圖示

首先系統會整合傳送進來每一筆主奏的音符群，一組 MIDI 訊息傳入到結束前，要先累積每一個音符的時間間隔時間，演奏者必須在下一組主奏音符群彈奏前，結束目前的主奏音符群。所謂「間隔時間」指的是鋼琴鍵從「按下」至「放開」的時間。

2.2.3 客制化的基本節奏、學習樣本之累積、伴奏修正微調機制、自動彈奏音符檢測

所謂客制化的基本節奏是每一次在累積學習樣本結束的時候，系統會依照不同演奏者的習慣不

同，量身定作出不同的基本節奏時間，而且也會讓原本固定標準樂譜間隔時間因不同使用者變成可變的，並且系統會自動詢問演奏者是否要改變基本節奏時間，此時再配合我們所定的硬體控制碼，所以如果以 1/4 拍為最小的拍數，1 拍等於四個 1/4 拍，將衍生下面的式子。

$$\text{總拍數} = \text{總節數} * 4 \text{ 拍} * 4 \quad (2)$$

$$\text{基本節奏時間} = \text{總花費時間} / \text{總拍數} \quad (3)$$

$$\text{可動的間隔時間} = \text{基本節奏時間} * \text{Tab} \quad (4)$$

由於本系統採取類神經網路實現伴奏系統的學習，所以學習樣本的正確性及累積顯得非常重要。首先，讓使用者彈奏主奏並聆聽伴奏來演奏一首曲子，此時主奏的 MIDI 訊息送進來，系統就開始分析此次送進來主奏的音符群是不是我們符合我們之前所設定的標準，若是，則依次把所有主奏的學習樣本全部都累積完畢。

而我們學習樣本的內容和以後我們測試時的輸入是採用「可動的間隔時間」減去「實際累積的間隔時間」(目前主奏音符群和前一個主奏音符群的間隔時間)，而得到「間隔時間差」。

所謂伴奏修正微調機制就是我們的類神經網路是以誤差決定誤差，系統將預測出下一筆尚未演奏之伴奏音的撥放時間，但是有時候演奏者可能針對某一個伴奏的間隔的時間提前修正太多，影響到前一個不相關的伴奏間隔時間，此時就必須進行所謂的伴奏修正微調機制，如果某一個伴奏被提前修正超原本的標準間隔時間時，就必須把修正過頭的間隔時間和標準間隔時間之間的差，平均的分給目前此次音符群前和上一次音符群之間的伴奏，讓他們間隔時間都提前修正一點，這樣的話既不會影響原本使用者想要的修正，更能使整體的伴奏聽起來更舒服一點!!

以圖 13 所示為例，假設系統預測出第二組音符群(主奏 2)相對應的伴奏(伴奏 3)必須向前修正 0.5 秒，但是該伴奏(伴奏 3)和前一個伴奏(伴奏 2)之間隔時間只有 0.3 秒，此時系統將會把原本要修正的伴奏(伴奏 3)只往前修正 0.3 秒，把本來要修 0.5 秒的差 0.2 秒，平均分給前面二個伴奏(伴奏 1、2)，讓它們都提早 0.1 秒撥放。



圖 13. 伴奏修正微調機制

此外，人們在演奏時難免會彈錯幾個音，所以

本系統加入自動檢查彈錯音的功能，告訴演奏者彈錯音在樂曲中所在的行、節、順序，清楚的讓演奏者知道。

以上的設計完全基於人性作考量，所以人性化和客制化的系統，將使得使用者操作起來更加順手，類神經網路學習結果的好壞取決先前學習樣本的準確性和樣本數量的多寡；學習樣本的數量越多，數據越正確，學習效果將會更佳。尤其對於初學者，若要使用模式 2 的伴奏效果，多次演奏輸入學習樣本，除了可以使自己的演奏實力提升，亦可讓學習樣本更準確，達成良好的智慧型伴奏效果。

2.2.4 類神經網路的伴奏學習

圖 14 是本系統所採用的類神經網路架構，包含了輸入層、隱藏層及輸出層，原本參考了模糊類神經網路的架構[10-12]，後來為求簡化，直接以類神經網路來實現。我們參考文獻[1]所提供的 C 語言程式，再根據本系統之需求，改以 VB 程式語言來實現。類神經網路運作過程，分為學習與回想兩部分，其學習的過程是一種監督式的學習從問題領域中取的學習樣本和期望值，並且將學習樣本輸入至網路中，反覆的調節網路連結的加權值；回想過程，則是一種驗證、分類或預測的過程。本系統則區分為「學習過程」以及「測試過程」兩部分，分別說明如下。

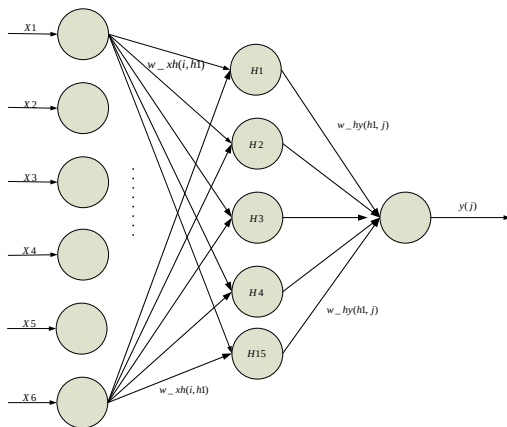


圖 14.類神經網路架構圖

學習過程方面：

有類神經網路學習經驗的工程人員應該都知道，類神經的網路架構與所要學習的系統息息相關，結構太大或是太小，都不適宜。經過許多測試，本論文的類神經網路的架構使用三層（多層）的網路架構，分別有六個輸入、十五個神經元構成的隱藏層，以及一個輸出。我們將前述所紀錄的學習樣本中每一次取出四筆間隔時間差，前三筆的間隔時間差當作是類神經網路的前三個輸入，第四筆的間

隔時間差則作為類神經網路的輸出期望值。類神經網路的第四個輸入，則是取當時樂譜的行數而第五個輸入則是取當時該樂譜節數而第六個輸入則是取當時該樂譜節數裡面的第幾個音符所合成的資訊，整理如下：

輸入 1：前二個音符群的間隔時間差
輸入 2：前一個音符群的間隔時間差
輸入 3：目前音符群的間隔時間差
輸入 4：行數
輸入 5：節數
輸入 6：個數

此點為本系統的特色，因為樂譜中的每一音符間的時間間隔，與其所在的音節息息相關，實驗驗證發現，加上這些輸入，可以大大的改善學習的結果。例如觀察過去文獻中的結果[4]，對於彈奏者的分析，平均就約有 0.1 秒的時間差，但對於專業的音樂家而言，他們則希望誤差在 0.05 秒以內。

但由於過去文獻的作者將整首樂曲一概而論，忽略了每一音節的演奏特性。過去文獻中的做法，是將目前所彈奏的音符群與標準樂譜間的時間差，以及前兩個音符群的時間差，總共三個值，作為系統的輸入。而事實上，彈過樂器的人都應該有所感覺，那就是對於無論初學者或是高級的演奏者而言，演奏的節奏特性，應該每一個音節都會不一樣的。例如對於初學者而言，某音節不熟練，所以那個音節的節奏反應自然會與其它地方大不相同；而對於熟練高級的演奏者而言，為了表達音樂的情感，也是在每一音節的詮釋都會自然地大異其趣的。

基於此，本系統在類神經網路的輸入上，所增加的第四、五、六個輸入節點，表示樂譜行數與節數和該節數的第幾個的資訊，以增加系統學習的成果，是非常具有意義的，結果也顯示可以大大地改善學習的成果。

由於類神經網路隱藏層的神經元是沒有一定的數目的，太多或是太少都不好，所以只能不斷的測試，圖 15 為系統隱藏層使用 5 個神經元的類神經網路收斂圖，而圖 16 為本系統使用 15 顆神經元的類神經網路收斂圖，由圖 15 和圖 16 兩個收斂圖的比較來說，一樣以學習次數 300 次而言，隱藏層使用 5 個神經元的系統只能收斂到 0.53，而隱藏層使用 15 個神經元的卻可以收斂到 0.37，很明顯的可以看出在本系統下使用隱藏層 15 顆神經元收斂效果比較好，至於說如果隱藏層如果使用超過 15 顆神經元時，有時反而會因為過度學習，而使收斂結果不佳，所以本系統的隱藏層採用 15 顆神經元。

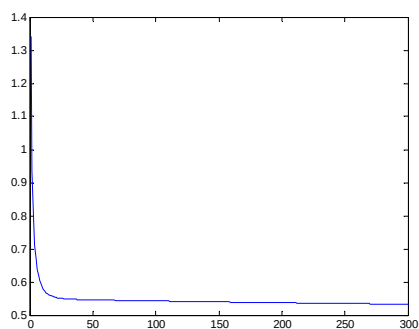


圖 15.系統收斂圖(隱藏層含 5 顆神經元)

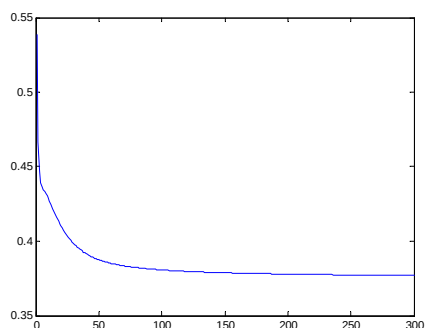


圖 16.系統收斂圖(隱藏層含 15 顆神經元)

而圖 17 所示為我們學習一組樣本的結果，而且學習樣本的時間差(實際累積的間隔時間和可量身定作的間隔時間誤差(4))。由於範例曲目共有 68 個音符群，故圖中橫軸為 68 次的輸入，每一次的輸入將預測出下一筆尚未彈奏出音符群相對應的伴奏提前或是延遲的修正時間，縱軸則為目標輸出值和實際預測輸出值之間的誤差(單位：秒)。學習後的結果，我們可以看到誤差少數幾筆在 0.06~0.1 之間，而也有少部份在 0.05~0.06 之間，但大部份都在 0.05 以下有些甚至更低，強調本系統具有智慧學習的能力。

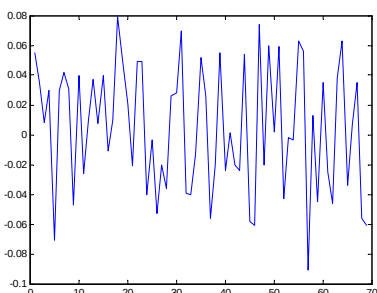


圖 17.一般誤差(學習結果)

測試過程方面:

在相同的類神經網路架構下，輸入的間隔時間變成即時資料，是以此測試程式方面，是結合「學習」以及「測試」兩大部分。在全部的測試完畢後，還會詢問使用者，對這一次的測試結果是否滿意？如果滿意的話，還可以重複學習這一次的即時輸入的測試樣本，更新類神經網路的加權值；但是如果不滿意的話，則將類神經網路的加權值還原為進入該次測試前的結果。

測試結果:

以圖 11 的鋼琴樂譜作為範例，基本上最後一筆學習樣本要等於即將測試的輸入或近似(演奏者久而習慣一定的節奏)。

但接下來我們將測試我們容許我們即時輸入樣本和最後一組的學習樣本有所出入，測試本系統是否能夠聰明的判斷出來我們要的結果，而圖 18 所示我們將容許即便減少即時的輸入樣本和最後一組學習樣本裡面每一組的時間差有 1%~5% 隨機變化(每筆間隔時間差都不一樣)，學習結果如圖所示。

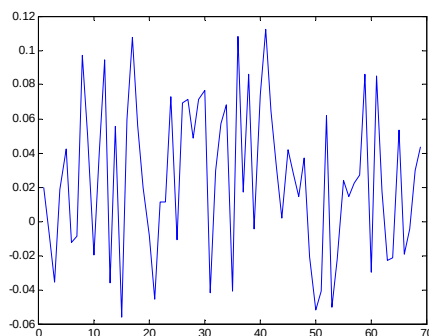


圖 18.一般誤差(變動幅度 1%~5%)

而且結果顯示只有少數幾筆資料在 0.1~0.12 之間，而有少部份在 0.05~0.1 之間，而大部份都在 0.05 以下

而圖 19 所示我們將容許即便減少即時的輸入樣本和最後一組學習樣本裡面每一組的時間差有 2%~10% 隨機變化(每筆間隔時間差都不一樣)，學習結果如圖所示。

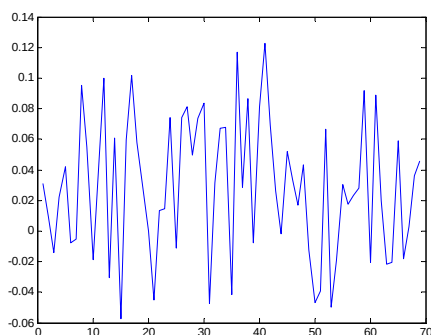


圖 19. 一般誤差(變動幅度 2%~10%)

而且結果顯示只有少數幾筆資料在 0.1~0.13 之間，而有少部份在 0.05~0.1 之間，而大部份都在 0.05 以下。

而圖 20 所示我們將容許即便減少即時的輸入樣本和最後一組學習樣本裡面每一組的時間差有 4%~20% 隨機變化(每筆間隔時間差都不一樣)，學習結果如圖所示。

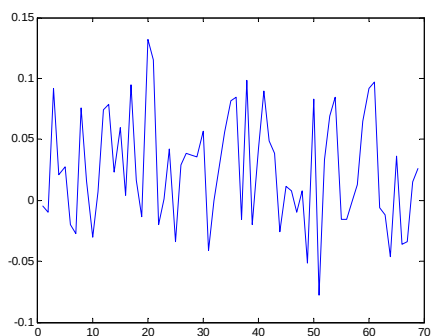


圖 20. 一般誤差(變動幅度 4%~20%)

結果顯示只有少數幾筆資料在 0.08~0.14 之間，而有少部份在 0.05~0.08 之間，而大部份都在 0.05 以下。

由以上的測試我們知道，在本論文下，雖然基本上系統測試時的即時輸入樣本要等於或是近似於最後一組的學習樣本，但是我們容許它們之間有所出入，而測試的結果發現減少的百分比愈大，會有少數幾筆的誤差變大，而其它的誤差雖然也有所變化，但是基本上還在容許範圍內。

結果顯示本系統除了能夠有智慧的學習之外，還能夠聰明的判斷不完整、有所變動輸入資料(有限的變動範圍內)，進而找出一組較佳的解答。在整體性能和人性化方面本系統已有明顯的提升，不論在系統本身的學習能力和抗雜訊能力以及人性化的互動都有不錯的表現。

科技的進步雖然帶給人們許多便利，實際上，仔細深思，人類付出的代價實在不小。如今所有的人們都走向了這條不歸路，如何將科技結合人文藝術，以求疏解人們的壓力及苦悶，此是科技人值得努力的方向。本論文結合了過去完成的自動吉他系統，應用了 MIDI 技術以及類神經網路學習技術等，實現了一個智慧型吉他演奏系統，同時也增進了自動演奏樂器的實用性、教學、與娛樂性。希望能帶動科技人員朝美化人生的方向邁進！

致謝

本論文承蒙行政院國科會計畫補助經費(計畫編號: NSC 95-2221-E-218-025)，得以順利執行完成，特此致謝！

參考文獻

- [1] 王進德，類神經網路與模糊控制理論入門與應用，全華。
- [2] 高盟欣，樂譜編輯與電腦音樂的應用，淡江大學資訊工程所碩士論文 1998。
- [3] 陳以哲，自動鋼琴之研製，交通大學控制工程所碩士論文 1991。
- [4] 張瑋娟，模糊類神經節奏控制器在自動伴奏系統上的應用，中華大學電機工程所碩士論文 2000。
- [5] 郭清界、張少華，卡爾卡西現代奏法 古典吉他大教本(上)(下)。台灣：天音出版社，2001。
- [6] 趙春棠，林奕志，進階自動吉他演奏系統之設計與實現，pp. d-197，2005 年中華民國自動控制研討會 2005。
- [7] 趙春棠，黃裕輝，一個實用的自動吉他樂曲輸入系統，2007 數位設計創意國際研討會，2007。
- [8] 溝淵浩五郎，卡爾卡西吉他教本，美樂出版
- [9] 蕭鈺恆，MIDI 在吉他編曲上的應用，淡江大學資訊工程所碩士論文 1998。
- [10] C.T. Chao and C.C. Teng, "Implementation of a fuzzy inference system using a normalized fuzzy neural network," *Fuzzy Sets and Systems*, vol.75, no.1, pp. 17-31, October 1995.
- [11] C.T. Chao, Y.J. Chen and C.C. Teng, "Simplification of fuzzy-neural systems using similarity analysis," *IEEE Trans. Systems Man Cybernet Part B: Cybernetics*, vol.26, no.2, pp. 344-354, April 1996.
- [12] C. T. Lin and C. S. G. Lee, Neural-network-based fuzzy logic control and decision system, *IEEE Trans. Computer C-40*(12) 1320-136, 1991
- [13] Mabry MIDI Software and Visual Basic Code: <http://home.brisnet.org.au/~mlevoi/midi.html>