

**Synchronous analyzing and
transforming for Guitar signals
composing**
即時吉他音訊處理及轉譜

學生：吳德霖、楊翔云、劉宇軒、胡祐瑄

指導教授：劉奕汶

國立清華大學電機工程學系

一、摘要

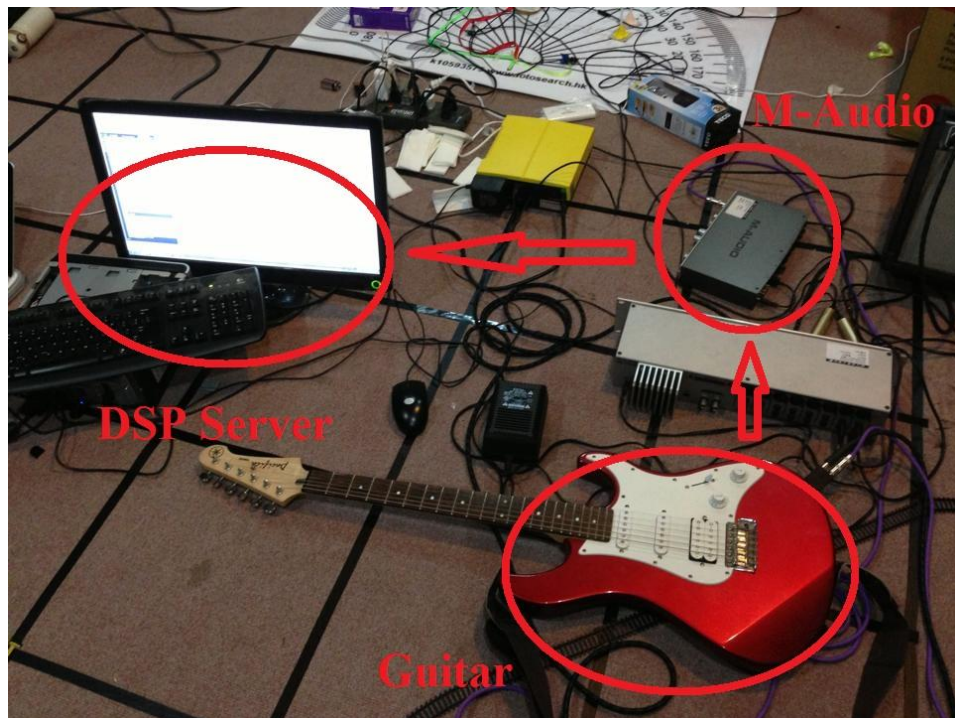
運用 Visual studio C#結合 M-audio(音訊介面卡)，將接收到的吉他音訊利用 Matlab 的 spectrogram 功能及其他相應函式對資料加以分析，並用 C code 將其轉換成一般符合 Guitar Pro(樂譜呈現程式)格式的輸入檔，讓音樂創作更輕鬆便利。

二、前言

一般樂手在音樂創作或是即興彈奏的時候，是用手寫在空白的五線譜上，即使有 Overture 或是 Guitar Pro 等輔助軟體，也沒有辦法做即時記錄，透過這個 project，我們可以讓音樂創作變得更簡單更有效率。當我們在彈奏時，便可以將彈奏的音符轉換成樂譜，甚至連看不懂五線譜的人也可以藉此擁有創作記譜的可能。另外，在練團室這種閉鎖音場中使用也不會被其他的樂器干擾，樂手也可以透過彈奏出來的樂譜與原曲比對並校正。

三、方法

硬體設備：

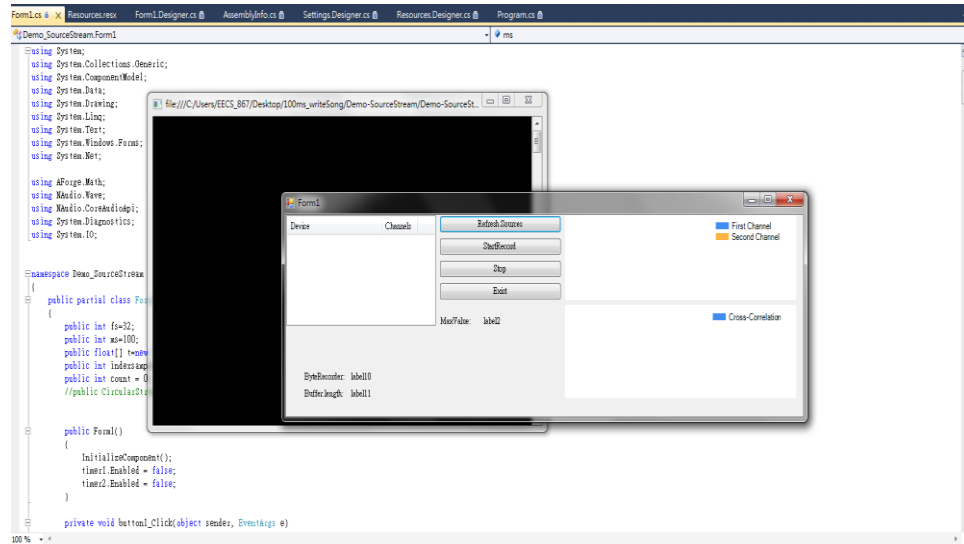


圖(1.1) 由上圖可知我們大略的架構及訊號傳輸流程，首先由吉他經由導線傳至 M-Audio 音效卡，並由 IEEE1394-a 火線傳至電腦主機，以 visual studio 為主程式進行 DSP 處理。

軟體層面(Coding part)分為以下三大部分：

1. Visual studio

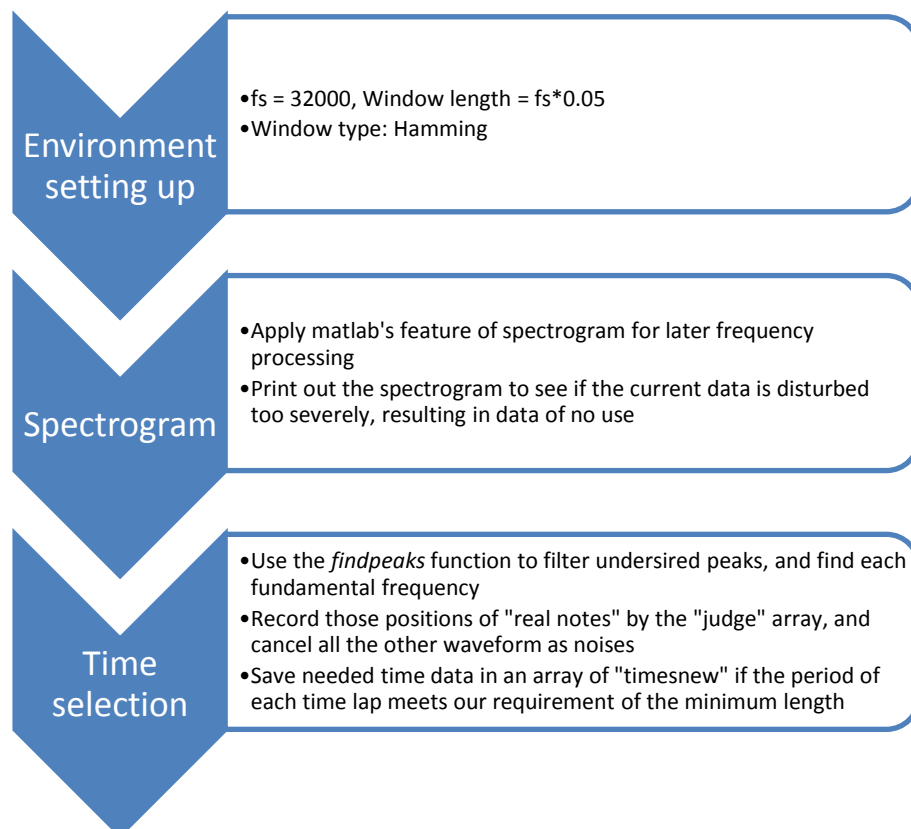
從孔學長那邊得到能夠處理 **real time** 訊號的 **C#**原始碼，經過嘗試後對此 **code** 進行 **sampling rate** 的修改，還有將原本雙 **channel** 改成單一 **channel**，我們便可用來讀取吉他訊號並轉換成 **floating point** 以便我們之後做訊號處理。

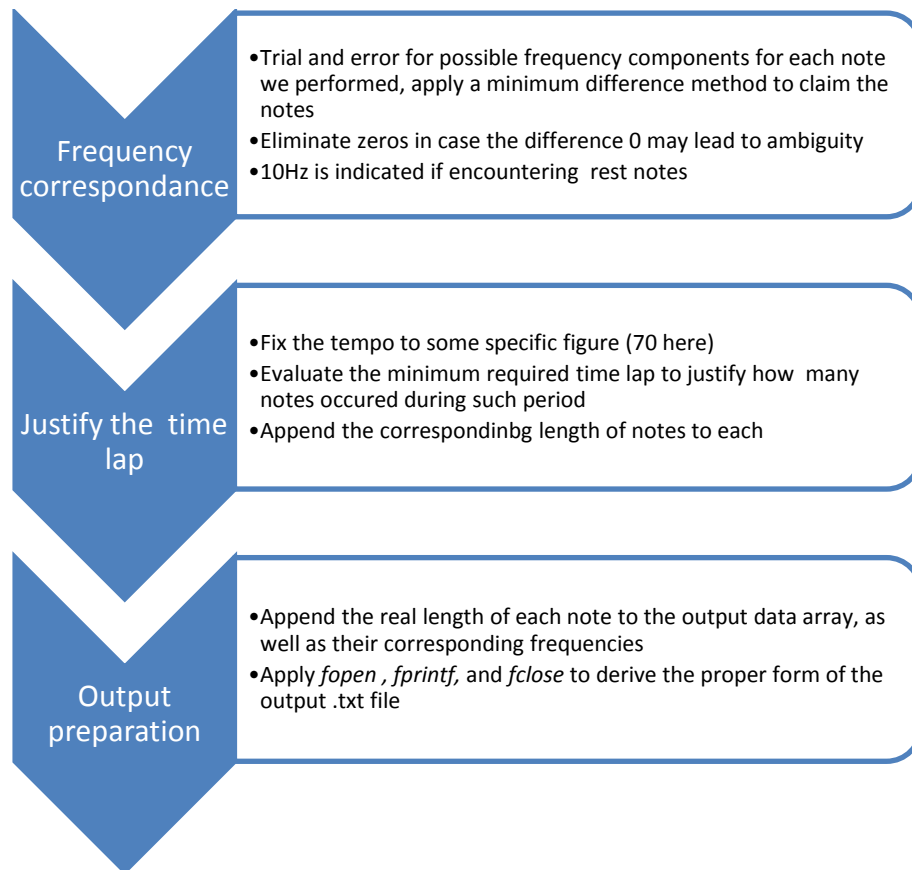


圖(1.2) 此圖為 **code** 所產生的圖形介面，經由此介面與硬體設備連結，來接收吉他 **real time** 的 **input**。

2. Matlab Coding

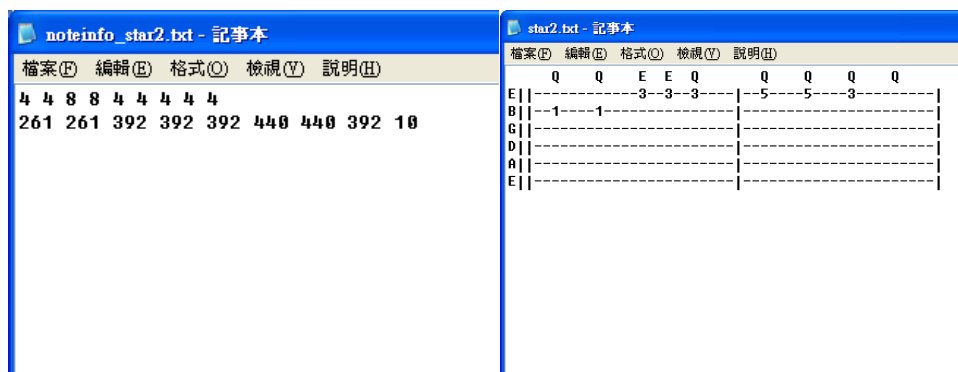
得到 Visual Studio 的測資後，以 Matlab 進行如下流程處理：



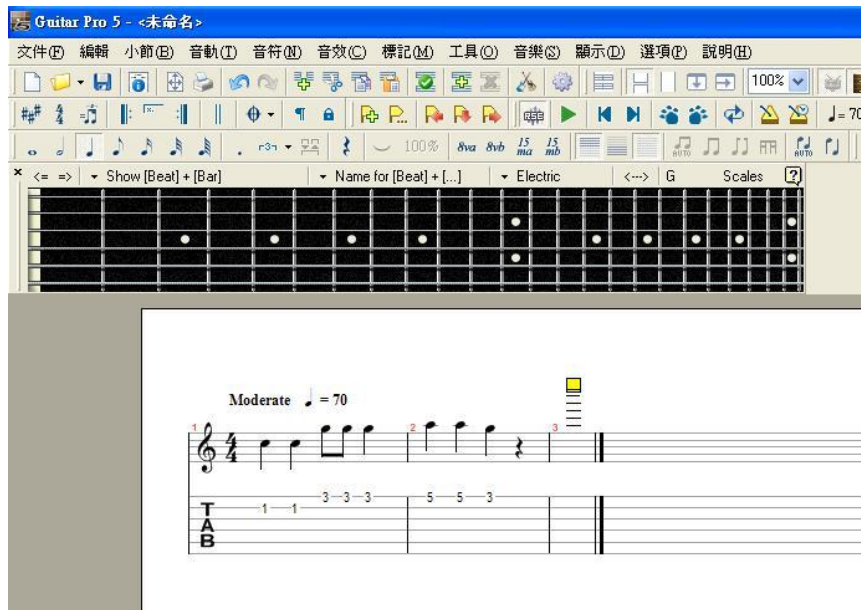


3.轉換成 Guitar Pro 譜

透過 C 語言，將從 Matlab 得到的 output.txt 檔之中含有音符節拍以及頻率的資料讀入，如圖(3.1)，我們的演算法可將 output 檔寫成 Guitar Pro 相容的 ASCII 導入檔案，如圖(3.2)。比方說，如圖(3.1)的 data，讀到的第一個音是(4,261)，4 代表的是四分音符，261 代表 D4 的頻率，轉換後就如圖(3.2)中所寫，第一個音為 Q (四分音符)，是按在吉他第二弦的第一個把位(D4)。將檔案導入 GP，就得到如圖(3.3)的結果，接著就可透過 GP 的播放功能，與原曲比較是否相符。



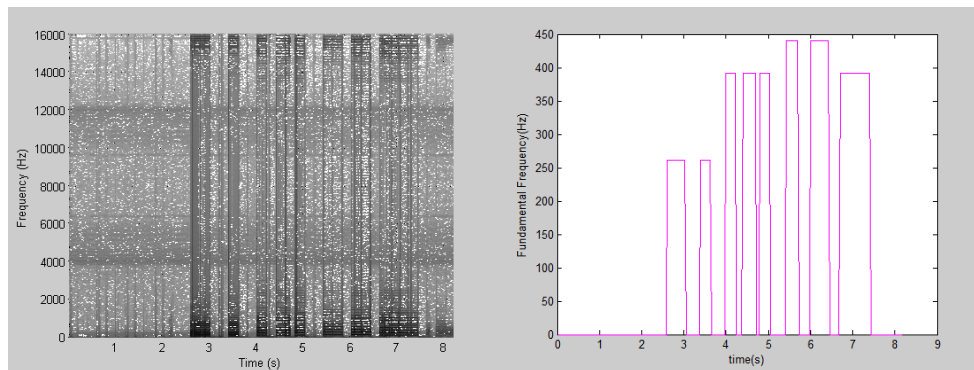
圖(3.1)(3.2) 左圖為 Matlab 輸出的.txt 檔，右圖為將左圖的.txt 檔經過 C 語言轉換後的檔案。



圖(3.3) 上圖為將圖(3.2)的.txt 檔匯入 GP 後的結果。

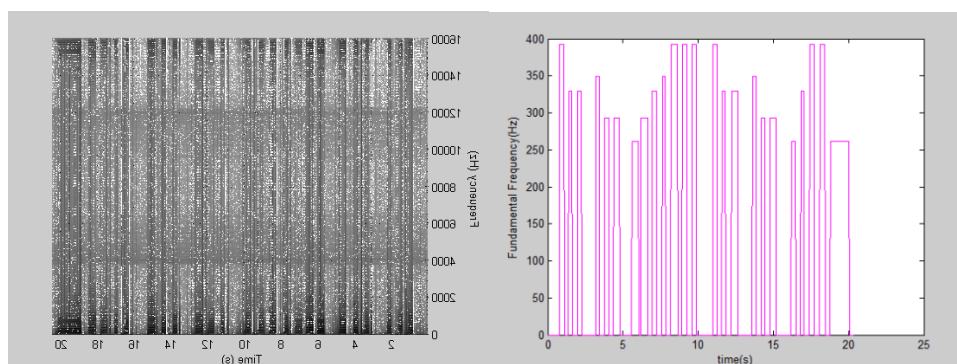
四、結果與討論

1.圖(4.1)(4.2)是我們其中一筆測試資料，可以看出每個音符相對節拍及頻率，將 Matlab 轉出的 output.txt 檔與演奏者所彈奏的樂曲比對之後，發現完全符合。接著經過轉檔程式(code 在附件檔案中)，得到如圖(3.2)的結果，此.txt 檔可以放入 Guitar Pro 中轉換成五線譜如圖(3.3)，播放後與原曲相同。

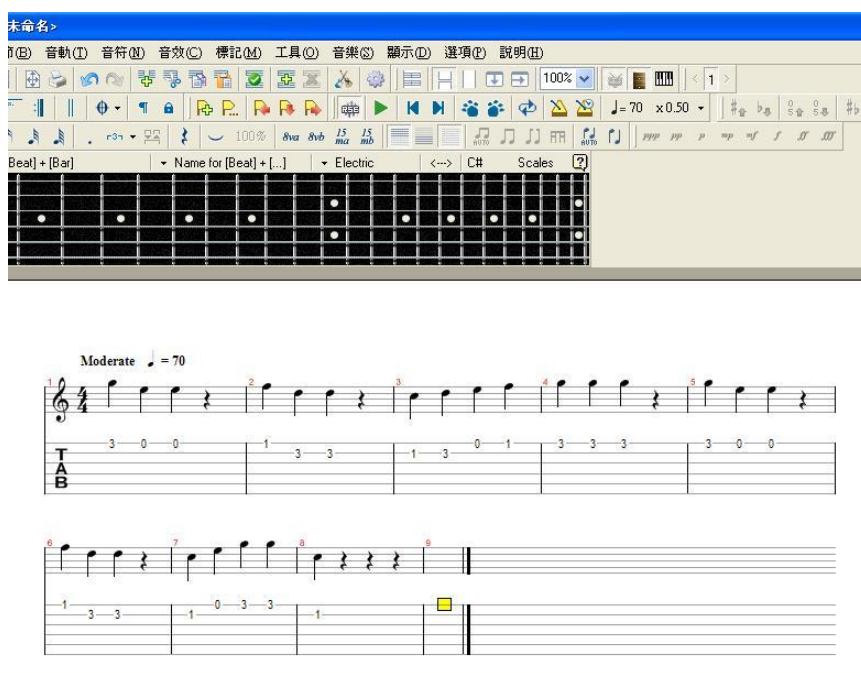


圖(4.1)(4.2) 此筆測資原曲為小星星的片段，用來測試可否分辨出四分與八分音符，左圖為其 Spectrogram，右圖為頻率對時間的分布圖。

2.圖(4.3)(4.4)是另一筆測試資料，由圖(4.5)轉檔後的樂譜可看出與預期相符合。



圖(4.3)(4.4) 此筆測資原曲為小蜜蜂的片段，用來測試可否分辨出四分音符與休止符，左圖為其 Spectrogram，右圖為頻率對時間的分布圖。



圖(4.5) 上圖為轉檔匯入 GP 後的結果

3.原先執著於直接使用 Visual studio 去處理 real time processing，以及相對應的 FFT，因為其處理過程過於繁雜，遇到很多問題，並常常必須引入額外的外嵌程式碼，變數之間的相容性及轉換在 C 類型語言也是個燙手山芋。Matlab 本身就有這些函式可供使用，其功能強大及完整，並且是直譯式語言，非常適合處理跟頻率相關的應用。因此放棄直接使用 Visual studio，而是以 Matlab 為訊號處理的主工具，這導致最後的結果並非真正的 real time processing，因為取節拍的演算法必須是一個段落

演奏完後做”事後”的即時處理，將讀取的訊號丟到 **Matlab** 做 **FFT** 以及 **Spectrogram** 和其他後續分析。因此，**Visual studio** 在本作中為一媒介角色，負責實現隔絕外界干擾以及未來展望中有機會達成真正的 **real time processing**。

4.一開始在構想 **project** 的時候為注重實用性，決定與 **GP** 軟體結合，遇到一個麻煩就是我們查詢不到 **GP** 譜中 **ASCII** 轉換的格式，所以我們花了不少時間才找出其規則，並撰寫相對應的 **C code** 去實現 **data** 間的相容性及可轉換性。

5.在架設硬體時，因為現在的筆電都沒有可以插音效擴充卡的插槽，後來發現 **Mac book** 系列有 **1394b** 的火線，就先購買了一條 **1394b** 轉 **1394a** 的線，結果因為 **Mac** 的 **virtual box** 並不支援火線的連接，而 **Visual studio** 又因不支援 **Mac** 得用虛擬主機開啟使用，我們只好放棄這個方法，最後是買了一張擴充卡插在 **PC** 上面連接吉他導線，才終於架設好硬體環境。

6.吉他頻率重疊(**replicate**)問題，這是完成這次 **project** 當中相對最棘手的困難點，舉例來說當演奏者彈奏的樂曲包含連續的 **D4** 和 **G4** 時，其波形圖含有的音頻成分可能同時包含 **261Hz** 和 **257Hz**，重複嘗試過多次後發現 **D4** 同時含有 **171Hz** 的頻率成分，我們於是把目標轉向類比 **257** 和 **171** 的差距。結果看似相當良好，也通過了多筆測資。不幸的是，在某種情況下這兩個音符在時域上的波形居然呈現兩個近乎完美且一模一樣的方波，使得音名判別上相對的困難許多。我們懷疑這是吉他弦本身的”弦外之音”，在某個特定音頻周期下可能由拾音器產生完全相同的基頻成分。

五、結論

因為硬體的限制，目前只能分析單音且節拍不超過 **90bpm** 的吉他訊號，若硬體環境能夠改善，將有可能可以處理即時快速的音訊處理，並用不完全是 **spectrogram** 分析的其他演算法處理和弦訊號，比如說直接從吉他拾音器讀取每一條弦的訊號並分離將會是最有可能及效率的方式。此外，如果我們用來訊號源不是用吉他，像是用 **keyboard** 的話，有比較乾淨的頻率，對於處理與分析會更準確。

在應用方面，目前樂手在創作或是在練團室練團，大多都還是只能靠手寫樂譜，這對於即興創作以及不熟悉五線譜的人來說都很不方便，也可能因未能即時手寫記譜失去很多靈感。此外，在吵雜的環境下錄音設備的干擾過大並且無法即時處理音訊，雜訊過大這項專案將失去意義，因

此我們認為，若此軟體開發持續發展，將很有可能具有實際的商業價值。值得一提的是，此次用到的分析方式，不只適用於吉他，整個樂的樂器包含鼓，貝斯等都能使用，因此有實現虛擬樂團以及雲端創作的可能性。

六、附錄

1. 音符/頻率對照表(吉他用)

<http://autsum.pixnet.net/blog/post/32948095-%E5%90%89%E4%BB%96%E5%90%84%E7%B5%83%E6%A0%BC%E7%9A%84%E9%9C%87%E7%9B%A%E9%A0%BB%E7%8E%87>

2. Visual studio tutorial clips

<http://www.youtube.com/watch?v=9PbNlOQEZfl>

3. 擴充卡



七、工作分配

組長(領導、分配)	吳德霖
Virtual studio C# code 修改	吳德霖、楊翔云
Matlab 分析	吳德霖、楊翔云
C code(matlab output.txt 轉成 GP 導入檔) 以及 分析操作 GP	劉宇軒、胡祐瑄
彈奏吉他提供測資	劉宇軒、胡祐瑄
硬體設置及操作	劉宇軒、胡祐瑄、吳德霖、楊翔云