

國立交通大學

工學院聲音與音樂創意科技碩士學位學程

碩士論文

以音高類集為理論基礎的自動作曲研究

A Study on the Pitch-Class Set Based Automated
Composition

研究生：柯舒方

指導教授：黃志方 教授

成維華 教授

中華民國一百年七月

以音高類集為理論基礎的自動作曲研究
A Study on the Pitch-Class Set Based Automated
Composition

研究生：柯舒方

Student : Shu-Fang Ko

指導教授：黃志方

Advisor : Chih- Fang Huang

成維華

Wei-Hua Chieng



Submitted to Master Program of Sound and Music Innovative Technologies
College of Engineering
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of Master
in
Engineering

July 2011

Hsinchu, Taiwan, Republic of China

中華民國一百年七月

以音高類集為理論基礎的自動作曲研究

學生：柯舒方

指導教授：黃志方

成維華

國立交通大學聲音與音樂創意科技碩士學位學程

摘要

本論文為以音高類集為理論基礎的自動作曲研究，自動作曲是利用電腦產生符合樂理規則的音樂，我們根據樂理或經驗來產生自動作曲的規則。在結構部分，加入動機變化與主題發展的功能，節奏產生則使用節奏複雜度來增加節奏的組合性，音高部分使用音高類集理論來創作，系統提供調性音樂及非調性音樂的樂曲產生，在調性音樂產生中，除了符合基本 24 個大小調的特性外，更加入終止式來加強調性音樂的和聲結構。本次研究的實驗結果，在調性辨識率上，調性辨識率平均為 78.25%，非調性的調性辨識率平均為 52.75%；在動機變化程度及喜好程度方面，我們使用 Likert 5-point Scale 的方法來計算分數，動機變化程度整體平均分數為 3.5，在水準之上；喜好程度比例依照調性音樂及非調性音樂各為 3.29 及 2.66，顯示大部分受試者對於調性音樂的接受度比較高。

關鍵字：自動作曲、演算法作曲、音高類集、節奏複雜度、動機與主題發展

A Study on the Pitch-Class Set Based Automated Composition

Student : Shu-Fang Ko

Advisor : Chih- Fang Huang

Wei-Hua Chieng

Master Program of Sound and Music Innovative Technologies
National Chiao Tung University

ABSTRACT

This thesis is based on Pitch Class Set theory to do the automated composition research. Automated composition can generate music applying music theory and rules with experiences into the generative music system. As to the music structure, motif variation and thematic development function is added. Rhythm generation can be performed by the rhythm combination with rhythm complexity. Pitch generation is based on the pitch class set theory to compose, and both tonal and atonal music patterns can be provided by the system automatically. Tonal music generation not only supports 24 major/minor keys, but also integrates the harmonic cadence implicitly. The experiment result shows the tonal music identification rate for the subjects is 78.25%, and the atonal music identification rate is 52.75%. With 5-point Likert scale, motive variation identification rate for listeners is 3.44 in average, and the generative music preference level is 3.29 and 2.66 for tonal and atonal music respectively, which shows that most people would prefer to accept tonal music, rather than atonal music.

Key words: Automated Composition, Algorithmic Composition, Pitch Class Set, Rhythm Complexity, Motif and Thematic Development

誌謝

轉眼間，兩年很快地過去了，很幸運可以進入這個特別的研究所，在這裡，除了課業上的學習外，還認識了許多見多識廣的人，兩年下來也累積了不少表演的經驗。

這兩年中，首先要感謝指導教授黃志方老師，除了在研究上的幫助外，老師提供許多表演機會，讓我得以累積舞台經驗，而在人生規劃上，也給予我寶貴的建議；再來要感謝曾毓忠老師在電腦音樂及科技樂器上的教導，讓我從中獲得不少專業知識。

感謝實驗室的同學、學長姐及學弟們，向斌學長在研究方面總是給予正確的建議，怡瑩學姊則是熱心大方，很照顧我們，毅凡學長是大家的好榜樣，是學弟妹學習的對象；感謝 molly (明瑾)、朱朱 (季芳)、粘粘 (為博)、囧豪 (志豪) 及志翰，可以和你們一起學習是一件很棒的事，也感謝學弟們，讓實驗室的氣氛更加歡樂。

最後感謝我的家人，從小一路支持我學習音樂，給予我無憂無慮的生活，也支持我進入這個研究所就讀，並讓我完成碩士學業，父母的教導讓我認真負責做好每件事，在此謝謝他們。



目錄

| | |
|---|-----|
| 摘要..... | i |
| Abstract..... | ii |
| 誌謝..... | iii |
| 目錄..... | iv |
| 表目錄..... | vi |
| 圖目錄..... | vii |
| 第一章、 概論..... | 1 |
| 1.1 動機..... | 1 |
| 1.2 背景知識..... | 1 |
| 1.3 論文架構..... | 3 |
| 第二章、 相關研究..... | 4 |
| 2.1 MIDI..... | 4 |
| 2.2 Pitch Class..... | 5 |
| 2.3 Pitch-Class Set..... | 6 |
| 2.3.1 Operations on Pitch Classes and Pitch Class Sets..... | 7 |
| 2.3.2 Normal Form..... | 9 |
| 2.3.3 Prime Form..... | 9 |
| 2.4 動機與主題發展..... | 10 |
| 2.4.1 動機..... | 10 |
| 2.4.2 動機處理..... | 11 |
| 2.4.3 主題..... | 13 |
| 2.4.4 主題發展..... | 13 |
| 2.5 節奏複雜度..... | 14 |
| 2.5.1 LHL (Longuet-Higgins and Lee Complexity)..... | 14 |
| 2.6 自動作曲..... | 15 |
| 2.6.1 Melody Generator..... | 15 |
| 2.6.2 The Algorithmic Composer: ALICE..... | 17 |
| 第三章、 研究方法..... | 18 |
| 3.1 系統架構..... | 18 |
| 3.2 找出 MIDI 音樂的 Prime Form..... | 20 |
| 3.3 節奏產生方法..... | 20 |

| | | |
|-------|------------------------|----|
| 3.3.1 | 3/4 拍節奏產生方式..... | 20 |
| 3.3.2 | 4/4 拍節奏產生方式..... | 21 |
| 3.4 | 音高產生..... | 22 |
| 3.4.1 | 非調性..... | 23 |
| 3.4.1 | 調性..... | 24 |
| 3.4.3 | Prime form filter..... | 25 |
| 3.4.4 | 終止式 (Cadence)..... | 27 |
| 3.5 | 動機及主題發展..... | 28 |
| 3.5.1 | 動機..... | 28 |
| 3.5.2 | 主題..... | 29 |
| 3.5.3 | 動機變化..... | 29 |
| 3.5.4 | 主題發展..... | 30 |
| 第四章、 | 系統功能與實作..... | 31 |
| 4.1 | 系統功能..... | 31 |
| 4.2 | 實作結果..... | 35 |
| 4.2.1 | Atonal 實作結果..... | 35 |
| 4.2.2 | Tonal 實作結果..... | 35 |
| 第五章、 | 實驗..... | 37 |
| 第六章、 | 結論與未來研究..... | 41 |
| 參考文獻 | | 42 |
| 附錄 | | 44 |

表目錄

| | | |
|-------|---------------------------------------|----|
| 表 2.1 | Header chunk 格式..... | 4 |
| 表 2.2 | Track chunk 格式..... | 4 |
| 表 2.3 | Pitch Class 對應表..... | 5 |
| 表 2.4 | 十一個音程對應表..... | 6 |
| 表 2.5 | Pitch Class 及 Inversion 對照表..... | 8 |
| 表 3.1 | 系統內建的 3/4 拍節奏樣本資料庫..... | 21 |
| 表 3.2 | 複雜度組合..... | 22 |
| 表 3.3 | 小節頭尾加音組合表..... | 23 |
| 表 3.4 | Prime form 所有變化對照表 (以[0,3,7]為例)..... | 24 |
| 表 3.5 | 大小調音階和 Pitch Class 對照表..... | 25 |
| 表 5.1 | 八首 MIDI 檔的調性辨識率統計表..... | 38 |
| 表 5.2 | 未受過音樂訓練受試者的八首 MIDI 檔的調性辨識率統計表..... | 39 |
| 表 5.3 | 八首 MIDI 檔的動機變化明顯程度統計表..... | 39 |
| 表 5.4 | 未受過音樂訓練受試者的八首 MIDI 檔的動機變化明顯程度統計表..... | 40 |
| 表 5.5 | 八首 MIDI 檔的喜好程度統計表..... | 40 |

圖目錄

| | | |
|--------|---|----|
| 圖 1.1 | 七種通用音符..... | 1 |
| 圖 1.2 | 音符長度關係圖..... | 2 |
| 圖 1.3 | C 大調自然音階..... | 2 |
| 圖 2.1 | mod 12 系統..... | 5 |
| 圖 2.2 | Pitch-Class Set..... | 6 |
| 圖 2.3 | Pitch-Class Set Transposition..... | 7 |
| 圖 2.4 | Pitch-class set inversion..... | 8 |
| 圖 2.5 | Normal Form..... | 9 |
| 圖 1.5 | 動機 (Beethoven's Fifth Symphony)..... | 10 |
| 圖 2.6 | Exact Repetition..... | 10 |
| 圖 2.7 | Modified Repetition..... | 10 |
| 圖 2.8 | Repeating..... | 11 |
| 圖 2.9 | Sequence..... | 11 |
| 圖 2.10 | Alternation..... | 11 |
| 圖 2.11 | Inversion..... | 12 |
| 圖 2.12 | Retrograde..... | 12 |
| 圖 2.13 | Augmentation (增值) 與 Diminution (減值)的範例..... | 12 |
| 圖 2.14 | 主題..... | 13 |
| 圖 2.15 | 主題發展..... | 13 |
| 圖 2.16 | LHL 分析節奏複雜度之樹狀圖..... | 14 |
| 圖 2.17 | LHL 複雜度分析例子..... | 15 |
| 圖 2.18 | Melody Generator 程式介面..... | 16 |
| 圖 2.19 | Melody Generator 產生旋律步驟圖..... | 16 |
| 圖 2.20 | ALICE 作曲流程圖..... | 17 |
| 圖 3.1 | 系統流程圖..... | 19 |
| 圖 3.2 | 「櫻花」譜例..... | 20 |
| 圖 3.3 | LHL 反推法之樹狀圖 (4/4, 16-beat)..... | 21 |
| 圖 3.4 | Prime form tonal filter 示意圖..... | 26 |
| 圖 3.5 | 正格終止 (Authentic cadence)..... | 27 |
| 圖 3.6 | 變格終止 (Plagal cadence)..... | 27 |

| | | |
|--------|--|----|
| 圖 3.7 | 終止式 $I \rightarrow IV \rightarrow V \rightarrow I$ | 27 |
| 圖 3.8 | 終止式 $I \rightarrow II^6 \rightarrow V \rightarrow I$ | 28 |
| 圖 3.9 | 主題與動機..... | 28 |
| 圖 3.10 | 系統主題架構..... | 29 |
| 圖 3.11 | 系統採用的主題發展方法..... | 30 |
| 圖 4.1 | 系統介面圖..... | 31 |
| 圖 4.2 | 讀入 MIDI File..... | 32 |
| 圖 4.3 | Prime Form 選擇..... | 32 |
| 圖 4.4 | Tonal 音樂選擇..... | 33 |
| 圖 4.5 | Atonal 實作結果..... | 36 |
| 圖 4.6 | Tonal 實作結果..... | 36 |
| 圖 5.1 | 受試者受過音樂訓練時間比例圖..... | 38 |
| 圖 5.2 | 受試者接觸過非調性音樂比例圖..... | 38 |



第一章、概論

1.1 動機

隨著電腦科技的進步，使用電腦代替創作、編曲、配伴奏等的演算法技術也迅速發展。目前電腦音樂的研究主題相當廣泛，除了上述電腦自動創作研究外，還包括音樂資訊檢索、音樂樣式探勘、音樂結構及和弦分析、音樂風格分類、音樂個人化及推薦等。

使用演算法來產生音樂已發展數十年之久，其中包括馬可夫鍊 [1, 2, 3]，以基因演算法為基礎的 evolutionary 方法 [4, 5] 等，另外有些以演算法作曲技術及樂理為基礎的旋律產生器，則提供專門的介面使產生調性旋律簡單化 [6, 7]。

而本系統的主要目的即為利用演算法來創作樂曲，我們以現代音樂中的音高類集為理論基礎，希望以此創作出多元化的旋律，系統加入以樂理為基礎的和聲及結構概念，另搭配節奏複雜度來產生變化豐富的節拍組合，藉以讓產生出的結果更有音樂性。

1.2 背景介紹

音樂隨著時間流動，若無記載的方法則無法保存及傳播音樂的內容，因此有了樂譜，即紀錄音樂的記號 [8]。樂譜是研究音樂的工具，其中，音符為樂譜的基本單位，代表樂音的符號，可表示音的長短，音符所在的位置可以表示音的高低。幾個音符會構成一個小節，許多小節構成完整的音樂。

音符本身所表示的長度為半數或倍數關係，通用的種類有七種，如圖 1.2 所示，全音符的長度為二分音符的兩倍，二分音符為四分音符的兩倍，圖 1.3 為各種音長之間的關係圖。



圖 1.1 七種通用音符

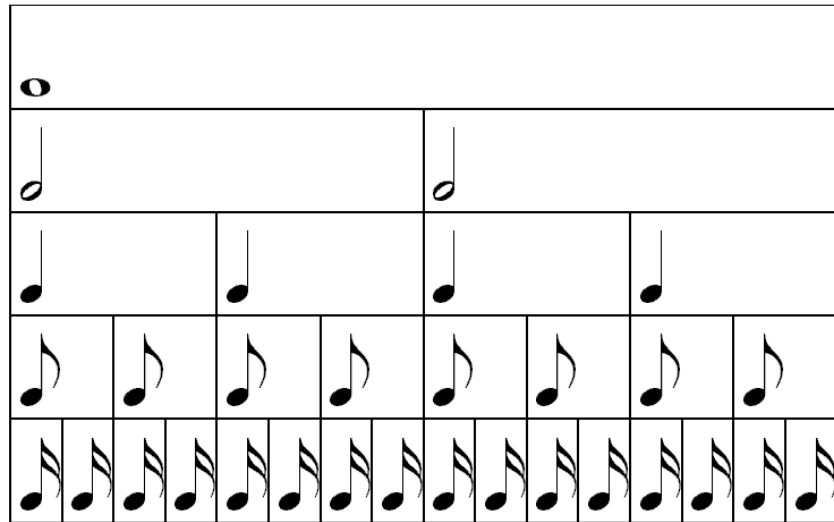


圖 1.2 音符長度關係圖

(資料來源：http://zh.wikipedia.org/wiki/File:YB0104_Valeur_relative_notes.png)

兩音高度上的距離為音程，半音為音程中最小的距離。音符按照其高低排列數個音的階級即稱為音階，表示音高的連續關係，曲調及和聲皆建立於音階之上。近代通用的音階為大音階及小音階，兩者皆屬於自然音階，由五個全音及兩個半音組成，兩者的差異為半音出現的位置，大音階的半音位置有二：一為第三音與第四音之間，另一個在第七音與第八音之間，小音階則依分類不同，半音出現位置也不同，圖 1.3 為 C 大調自然音階的範例。

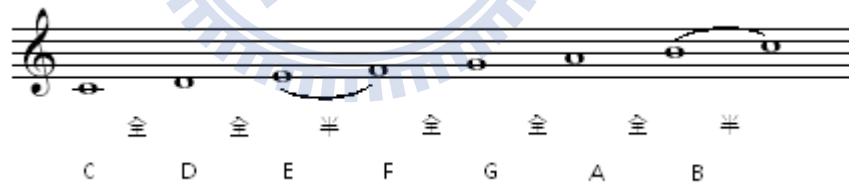


圖 1.3 C 大調自然音階

現代音樂家中，曾試用新音階創作音樂 [8]。新音階裡最著名的為全音階 (Whole-tone scale) 與十二聲音階 (Twelve-tone scale)。全音階全由全音組成，這種音階迴避了半音的傾向性，因而具有一種游移不定的獨特魅力。十二聲音階則是一個獨立的音階，建築於此音階之上的音樂便為無調音樂，無調音樂打破有調音樂和聲上的法則，創作的曲調由不協和的和絃組成，奧國作曲家荀白克為無調音樂的創始者。

就音樂的本質來看，音樂由許多元素所構成，而樂理為音樂的基礎，其包含許多規範，作曲家在作曲時的準則即建立在樂理之上，例如：調性、織體、曲式、和聲、旋律等。在調性音樂中，調性分為大小調共 24 個調，不同的調由不同的

音階組成，聽起來的感覺也不盡相同；織體與曲式為音樂的結構形式，織體為空間上的形式，曲式則是時間上的形式。一段時間內所聽到的聲響層次即為織體的密度大小，而音樂在時間上的延續，無論長短都必須有一個結構框架，即為曲式；和聲的主要功能是用陪襯旋律，不同的和聲會造成不同的效果；旋律則是由音高及節奏組成，為音樂中最重要要素，旋律具有其獨特的特徵，通常能讓聽者印象深刻。

1.3 論文架構

本論文以音高類集為理論基礎，加入各種樂理規則，及節奏複雜度概念，藉由音高類集豐富的變化及節奏複雜度多樣組合，來產生出各種不同的樂曲。

因此本論文在第二章相關研究中，先介紹各種樂理知識，包括 Pitch-Class Set、節奏複雜度、動機與主題發展等，另外還有講述自動作曲的發展。第三章為研究方法，我們介紹如何運用這些樂理加入系統中，來產生樂曲，並於第四章系統功能與實作中，展示我們開發的系統及產生的音樂。第五章為實驗，其中介紹我們的實驗設計、實驗評估及實驗結果，最後在第六章總結我們的結果，及討論未來研究可能的方向。



第二章、 相關研究

2.1 MIDI

現今數位音樂儲存的方式分為 Symbolic 及 Acoustic 兩種。Acoustic 將音樂儲存為聲音訊號，取樣頻率高低會影響播放效果的好壞，其包含的音訊資訊最為完整，但音樂的內容卻不易得知；而 Symbolic 則將音樂以不同事件來記錄，再用模擬的樂器聲音來播放，效果好壞取決於製作的水平及模擬音源器的聲音。MIDI 格式為最常見的 Symbolic 儲存方法。

為解決電子樂器間的通訊問題，1981 年 Dave Smith 提出 MIDI(Musical Instrument Digital Interface) 格式。由於 MIDI 格式是以事件儲存，因此不需透過音訊處理技巧就能擷取各種音樂參數，例如：音高 60 等同於中央 C，只要能正確解讀 MIDI 事件即能取得想要的結果。

常見的 MIDI 有三種不同的格式。一般最常見的格式為 GM(General MIDI)，另外兩種為 GS(General Standard)與 XG(Extended General MIDI)。其檔案的架構是由 Header chunk 與多個 Track chunk 所組成。Header chunk 主要紀錄播放格式、音軌數量、演奏的單位時間等資訊，如表 2.1。Track chunk 則記錄許多 MIDI 事件，如表 2.2，每個 MIDI 事件都會紀錄發生的時間。一個音符包含兩個 MIDI 事件：Onset 事件和 Offset 事件。Onset 事件代表音符開始發音，Offset 事件代表音符結束發音。兩個事件中都會記錄時間、音高、音長、力度、頻道等資訊。其中，音高的範圍是 0~127，以半音為單位；音名從 C0~C9，中央 C 在 C4，以數字表示是 60；音長則是 Onset 事件與同音高的 Offset 事件之時間間距。

表 2.1 Header chunk 格式

| Header chunk | | | | |
|--------------|-------------|----------|----------|------------|
| Chunk Type | Data Length | Data | | |
| 4 bytes | 4bytes | 6 bytes | | |
| | | 2 bytes | 2 bytes | 2 bytes |
| MThd | <length> | <format> | <tracks> | <division> |

表 2.2 Track chunk 格式

| Track chunk | | |
|-------------|-------------|---|
| Event Type | Data Length | Data |
| 4 bytes | 4bytes | <length> bytes |
| MTrk | <length> | <delta time><event>,<delta time><event>,... |

2.2 Pitch Class

在音樂中，每個 Pitch 都有各自精確的頻率，而 Pitch Class 指的是將一群同名 Pitch 分類為一組，它不是單獨的個體，是將同音高的 Pitch 以相隔八度或其倍數隔開，例如：Pitch Class A 包括所有名為 A 的 Pitch。

一般調性音樂裡， B^b 並不同於 A^\sharp 。由於最初音律是由五度相生律（五度圈）而成，是純律的物理和諧倍數關係，造成音與音間的音程有所差異，在轉調時會產生不悅耳的音律，因此有了十二平均律的概念。十二平均律將一八度音程等比例分成 12 等份，每等分為一個半音，因為每個音間隔相等且和純律相差不大，也就解決了轉調問題。雖然在調性系統裡， B^b 與 A^\sharp 被賦予的功能及意義不同，但因為使用了十二平均律，它們聽起來的音高會是差不多的，在 Pitch Class 裡則會用相同的代號代表，表 2.3 為 Pitch Class 及各個音高的對應表。

表 2.3 Pitch Class 對應表

| 音名 | C | C [#] D ^b | D | D [#] E ^b | E | F | F [#] G ^b | G | G [#] A ^b | A | A [#] B ^b | B |
|-------------|---|----------------------------------|---|----------------------------------|---|---|----------------------------------|---|----------------------------------|---|----------------------------------|----|
| Pitch class | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |

Pitch Class 將相差八度的 Pitch 歸為一類，以數字表示就是 0-11，若要將某個 Pitch 提高八度或降低八度，得到的 Pitch Class 仍是一樣的；例如想把 E^b 提高八度，則 Pitch Class 的計算方法為： $3 + 12 = 15$ ，然而 Pitch Class 是以 0-11 為一個循環（圖 2.1），因此 15 在 Pitch Class 裡等同於 3，也就是 mod 12 的系統。在 MIDI 中，取得音高後，將音高除以 12 取餘數，即能得知 Pitch Class，例如：音高 67 的 Pitch Class 為 $67 \bmod 12 = 7$ 。

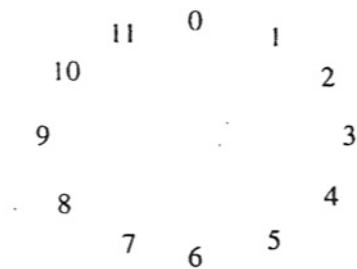


圖 2.1 mod 12 系統 [9]

由於異符同音的 Pitch 在 Pitch Class 裡視為相同，因此不同名稱但絕對大小相同的音程會被分為同類，例如，減四度及大三度，它們在音程的意義上即使不同，但絕對音程大小皆為四個半音差，所以利用 Pitch Class 的概念，兩者皆可由數字 4 表示。表 2.4 為十一個音程對應表。

表 2.4 十一個音程對應表 [9]

| | | | | | | | | | | | | |
|------|------|-----|-----|-----|-----|------|-----|------|-----|-----|-----|-----|
| 音程名稱 | 完全一度 | 小二度 | 大二度 | 小三度 | 大三度 | 增三度 | 增四度 | 完全五度 | 增五度 | 大六度 | 增六度 | 大七度 |
| | 完全八度 | | 減三度 | 增二度 | 減四度 | 完全四度 | 減五度 | 減六度 | 小六度 | 減七度 | 小七度 | |
| 對應數字 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |

2.3 Pitch-Class Set

Pitch Class 將所有音高以 12 為一組，以 0 至 11 表示，數字代表任一八度上的相同音高，即不考慮音域。例如：0 代表音高 C，7 代表音高 G，音高 60 和音高 72 皆是 C 音。而 Pitch-Class Set 則是一組無序的 Pitch-Class 集合，例如：[0,4,7]，它可利用 Transposition、Inversion 及 T_nI 作變化，主要用於分析 12 音列的作品。

圖 2.2 Pitch-Class Set
(Schoenberg, Gavotte from Suite for Piano, Op. 25) [9]

圖 2.2 是五個摘錄自荀白克第二十五號鋼琴組曲“Gavotte”的 Pitch-Class Set 的例子，圈起來的是一組 Pitch-Class Set(D^b, E, F, G)，這組 Pitch-Class Set 以不

同的表達方式在出現在音樂中，它以旋律形式出現在第一小節開頭 (a) 及第七小節 (b)；在第十六小節中開頭時，以雙音的形式出現，此時為樂曲的一半 (c)；到了第二十四小節時改用和絃 (d)；最後到了二十七小節結尾時又重新出現，代表此曲到最後仍使用此 Pitch-Class Set 為音樂想法 (e)。

2.3.1 Operations on Pitch Classes and Pitch Class Sets

● Transposition

Transposition 是將 Pitch-Class Set 內的所有數字作加(或減)的運算，以 T_n 表示，例如： $[2,3,7] = T_2[0,1,5]$ 。若運算後的數字大於 12，則須取 mod 12 後的數字，例如： $[11,0,4] = T_{11}[0,1,5]$ 。

圖 2.3 為 Pitch-Class Set Transposition 的例子，圖中被圈起來的四處為 Pitch-Class Set Transposition 的地方，) 的 Pitch-Class Set 為 $[3,4,7]$ ，(為 $[2,3,6]$ ，) 是 $[11,0,3]$ ，(為 $[8,9,0]$ ，其中， $[3,4,7] = T_1[2,3,6]$ ， $[3,4,7] = T_4[11,0,3]$ ， $[2,3,6] = T_3[11,0,3]$ ， $[11,0,3] = T_3[8,9,0]$ 。

圖 2.3 Pitch-Class Set Transposition
(Webern, Concerto for Nine Instruments, Op. 24)[9]

● **Inversion**

Inversion 是用 12 去減掉 Pitch-Class Set 內的所有數字，以 I 表示，例如： $[0,7,11] = T_0I[0,1,5]$ 。表 2.5 為 Pitch Class 及 Inversion 的對照表。

表 2.5 Pitch Class 及 Inversion 對照表

| pitch class(n) | Inversion (12-n) |
|----------------|------------------|
| 0 | 0 |
| 1 | 11 |
| 2 | 10 |
| 3 | 9 |
| 4 | 8 |
| 5 | 7 |
| 6 | 6 |
| 7 | 5 |
| 8 | 4 |
| 9 | 3 |
| 10 | 2 |
| 11 | 1 |

圖 2.4 為 Pitch-Class Set Inversion 的例子，圖中被圈起來的前二處為 Pitch-Class Set Inversion 的地方，) 的 Pitch-Class Set 為 $[7,8,11]$ ，(為 $[1,4,5]$ ， $[1,4,5] = I[7,8,11]$ ，其中，7 的 Inversion 為 5，8 的 Inversion 為 4，11 的 Inversion 為 1。



圖 2.4 Pitch-class set inversion
(Schoenberg, Piano Piece, Op. 11)[9]

● **T_nI**

T_nI 為綜合 Transposition 及 Inversion 的變化，先將 Pitch-Class Set 作 Inversion 後再 Transposition，圖 2.4 中被圈起來的第一、三處為 Pitch-Class Set T_nI 的地方，) 的 Pitch-Class Set 為 $[7,8,11]$ ，) 為 $[6,9,10]$ ， $[6,9,10] = T_5I[7,8,11]$ ，其中， $[1,4,5] = I[7,8,11]$ ， $[6,9,10] = T_5[1,4,5]$ 。

2.3.2 Normal Form

一個 Pitch-Class Set 可用不同變化形式出現在音樂裡，然而不論它出現的形式如何，皆可被整理成一個“Compact“的形式，也就是 Normal Form。Normal Form 是以 Compact Order 而存在的形式，它表示此 Pitch-Class Set 裡，頭尾 Pitch Class 間 Interval 的值為最小。



圖 2.5 Normal Form

圖 2.5 為 Normal Form 的例子，Pitch-Class Set 組成音為 C、G[#]、B、E、A，首先計算頭尾 Pitch Class 間 Interval 的值，從圖中可看出（及）的 Interval 值為最小，而在排法）裡，G[#]及 C 間的 Interval 值比排法（小，因此排法） [8,9,11,0,4] 為 C、G[#]、B、E、A 的 Normal Form。

2.3.3 Prime Form [9, 10, 11]

Set Classes 有兩種標準的命名方法，第一種是使用由 Allen Forte 所設計的方法，它定義每種 set class 由一組數字代表，例如：3-4，3 表示 Pitch-Class Set 裡有幾個 Pitch Class，4 則是此 Set Class 在 Forte 所制定的列表裡的位置，此種方法被廣為運用；第二種方法則是從 Set Class 裡所有的成員中，選出最 Normal 的 Normal Form 來代表，即被稱為 Prime Form，Prime Form 皆從 Pitch Class 0 開始。每組特定的 Pitch-Class Set 都有一個獨特的 Prime Form，若兩組 set 的 Prime Form 相同，則表示它們包含相同數目的 Pitch 及音程，所以聽起來的感覺會是差不多的，例如：Pitch-Class Set [1,2,7]、[8,2,3] 及 [0,11,6] 的 Prime Form 皆為 [0,1,6]，則它們聽起來的聲響差異不大。

Prime Form 的計算方法為：找出此 Pitch-Class Set 的 Normal Form 及 Inverted Normal Form，接著比較二者由左至右 Interval 的 Packed 程度，選出 Packed 程度最大的一個，並將此 Normal Form 的第一個 Pitch Class 作 Transposition 至 0，即為 Prime Form。以 Pitch-Class Set [7,8,2,5] 為例，其 Normal Form 為 [2,5,7,8]，Inverted Normal Form 為 [4,5,7,10]，由於 Inverted Normal Form 的 Interval 為：1,2,3，Normal Form 的 Interval 為：3,2,1，經比較，1,2,3 的 packed 程度較 3,2,1 大，即 [4,5,7,10] 的 packed 程度較 [2,5,7,8] 大，因此將 [4,5,7,10] 的 4 作 Transposition 至 0，也就是 T₈，而得到 Prime Form 為 [0,1,3,6]。附錄為所有的 Prime Form 列表。

2.4 動機與主題發展

動機和主題是作曲的基本想法，動機包含了一組特有的音高結構及節奏樣式，是主題的一部分。主題是以動機發展而完成，不同主題構成不同段落，再由這些段落組成一首音樂。

2.4.1 動機

動機是音樂的基本單位，通常出現在作品的開頭，讓人留有印象且具特色 [12]。動機由音程和節奏組成，是曲調的出發點及要素，藉以構成樂節、分句及樂句。例如：貝多芬第五號交響曲“命運”第一樂章的主要動機，即最初的四個音“Sol Sol Sol Mi”(圖 2.5)，不斷在整個交響樂中的第一樂章出現。

The image shows a musical score for the first movement of Beethoven's Fifth Symphony. It features four staves: Violinen, Klarinetten (Violins, Clarinets), Violoncelli, Bassen (Violas), and Celli, Bässe (Cellos, Basses). The key signature is one flat (B-flat major/D minor) and the time signature is 2/4. The main motif is a four-note sequence: G4, G4, G4, F4. The first staff (Violinen, Klarinetten) starts with a treble clef and a 2/4 time signature. The second staff (Violoncelli, Bassen) starts with a bass clef and a 2/4 time signature. The third staff (Celli, Bässe) starts with a bass clef and a 2/4 time signature. The fourth staff (Celli, Bässe) starts with a bass clef and a 2/4 time signature. The motif is repeated in each staff. The dynamic marking *ff* is present in the second staff.

圖 2.5 動機 (Beethoven's Fifth Symphony)

(資料來源：<http://en.wikipedia.org/wiki/File:FuenfteDeckblatt.png>)

動機會藉由不同形式重複出現在整首曲子裡，以避免讓音樂只是單調的重複同一動機，通常重複形式有 Exact、Modified 或 Developed。“Exact”方法保留所有特徵及關係，例如：Transposition、Inversion、Retrograde、Retrograde Inversion、Diminutions 及 Augmentation (圖 2.6)。“Modified”方法則會改變原有動機的一些特徵及保留剩餘的部分 (圖 2.7)。

The image shows three examples of exact repetition of the motif from Figure 2.5. Example a) shows the motif in its original form, followed by its Inversion (mirrored across a horizontal axis), Retrograde (played backwards), and Retrograde Inversion (mirrored and played backwards). Example b) shows the motif with Diminution (shorter note values). Example c) shows the motif with Augmentation (longer note values).

圖 2.6 Exact Repetition

The image shows a single example of modified repetition of the motif. The motif is repeated, but with some notes altered in pitch or rhythm, illustrating how the original motif is changed while retaining some of its characteristics.

圖 2.7 Modified Repetition

2.4.2 動機處理

動機處理是描述作曲的程序，在此作曲的程序中，完整的作品或樂節是以主題的動機為依據，連續的動機結合形成半樂句，連續的半樂句結合而成樂句 [13]。以下介紹由 Stein [13, 14] 及 Schoenberg [12] 提出常用的六種動機處理變化特性：

- **Repetition**

Repetition 可以加強人們對於主題動機的印象，其帶出來的效果並非複製，而是平衡與對稱的感覺，例如孟德爾頌無言歌 No. 45 的開頭即用 Repetition 的手法 (圖 2.8)。



圖 2.8 Repeating
(Mendelssohn, Songs Without Words, No. 45)

- **Sequence(模進)**

Sequence 至少有兩次相同重複的動機，通常為單一行進方向：更高或者更低。而 Sequence 又分為兩種，一種是 Real Sequence，另一種是 Tonal Sequence。Real Sequence 不考慮調性，因此 Transposition 的關係是絕對的，Real Sequence 有時聽起來會使調性改變；Tonal Sequence 則是同一調性內部 Sequence，Transposition 時會將音程調整至正確的調性音階上。圖 2.9 的例子屬於 Tonal Sequence，Transposition 時會調整到 F 大調音階上，而第四小節則是 Modified Sequence。



Modified Sequence

圖 2.9 Sequence
(Beethoven, Symphony No. 6, first movement)

- **Alternation (交替)**

Alternation 為兩個動機交互出現，例如莫札特第五號奏鳴曲(圖 2.10)中，a 動機和 b 動機形成一個半樂句，各自做動機處理後，又結合形成另一個半樂句。

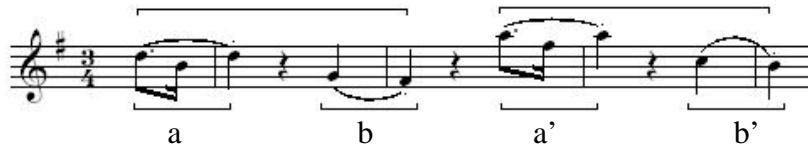


圖 2.10 Alternation (Mozart, Sonata No. 5, K. 189h)

- **Inversion**

Inversion 指的是原動機的反方向行進，例如 2.11 中，第二小節的動機 Contour 原本是 -, +, +, +，第三小節反向進行後，Contour 變成 +, -, -, -。Inversion 會保留原本動機的節奏，音程也會保留，但會照著行進方向走，因此原本是往下大二度，到了 Inversion 時會變成往上大二度。



圖 2.11 Inversion
(Stravinsky, The Firebird)

- **Retrograde(逆行)**

Retrograde 會維持動機節奏，但音高的次序相反，以蓋西文 "I Got Rhythm" 為例，音高原本是 "Fa Sol Si Do" 變為 "Do Si Sol Fa" (圖 2.12)。



圖 2.12 Retrograde
(Gershwin, I Got Rhythm)

- **Augmentation (增值) 及 Diminution (減值)**

Augmentation 將動機的節奏依比例加快，Diminution 將動機的節奏依比例減慢，兩者皆保留原本的音高。圖 2.13 中，上方為原始動機，下方前三小節為 Diminution，後面五小節則是 Augmentation。



圖 2.13 Augmentation (增值) 與 Diminution (減值) 的範例

2.4.3 主題

主題通常是一個完整的段落 [12]。在簡單的情況下，主題結構由偶數的小節數組成；最常被使用的是八或八的倍數。如果開頭是一段兩小節的樂句，則第三和第四小節有可能是前二小節的 Transposition Repetition，如圖 2.14 為海頓 G 大調奏鳴曲第一樂章的第一主題，第四小節至第六小節第一拍，為第一小節至第三小節第一拍的 Transposition Repetition。



圖 2.14 主題

(First theme Haydn's Sonata in G Major, Hob XVI: G1, l, mm. 1-12)

2.3.4 主題發展

主題發展 [15, 16] 包含一個較短的主題（或是多個較短的主題），利用 Transposition、Interval Expansion and Contraction、Rhythmic Augmentation and Diminution、Inversion、Tonality Changes 等方法來作主題發展 [15]。如圖 2.15，原始的主題 (a) 用不同的方法變化成 (b) Transposition、(c) Interval Expansion、(d) Rhythmic Augmentation、(e) Rhythmic Diminution 和 (f) Inversion。變化方法有很多種，但基本上不會破壞原本的主題特徵。

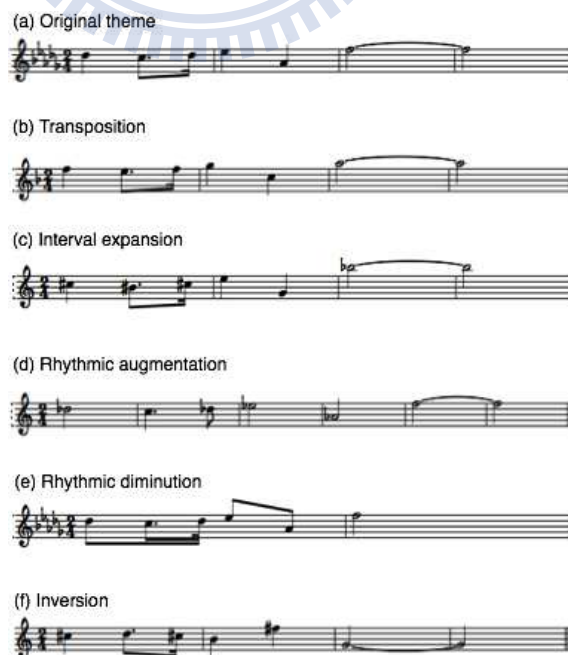


圖 2.15 主題發展 [15]

2.5 節奏複雜度 [17, 18]

複雜度是節奏的特徵之一，而對於人類感知節奏程度的方法中，以 LHL 分析法最為接近，LHL 計算人對節奏切分音(Syncopation)的感知程度。一般來說，複雜度越高，切分音出現越多，節奏聽起來就越複雜，反之亦然。

2.5.1 LHL (Longuet-Higgins and Lee Complexity) [19]

LHL 節奏複雜度分析法是由 H. Longuet-Higgins 和 C. Lee 所提出，在不同的節奏架構 (3/4、4/4 拍) 內，會有不同的權重表示法。他們用樹狀圖來表示 Metrical Hierarchy 及各個音符的權重，其他文獻中也提到更多有關 LHL 的細節及實作方法 [20, 21]。

LHL 是一種找切分音的方法，根據 Metrical Hierarchy 找出切分音出現在小節何處。LHL 調整 Metrical Hierarchy 的權重，0 是最高權重，其餘則遞減，LHL 根據小節內最小單位，將小節分成最小單位的數量等分，例如：4/4 的節奏架構裡，最小單位是 16 分音符，則分成 16 等分；若為 3/4，則分成 12 等分。等分後將架構畫成樹狀圖 (圖 2.16)，接著在各等分處標出各其權重，除以 1 之處為 0，除以 2 之處為 -1，除以 3 之處為 -2，除以 4 之處為 -3。

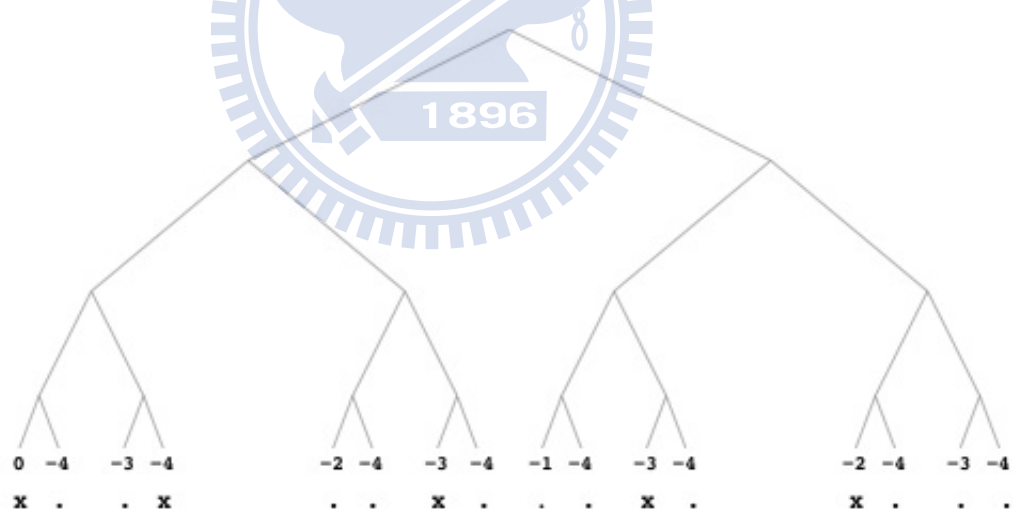


圖 2.16 LHL 分析節奏複雜度之樹狀圖 [22]

權重標完後會找出 Rest(.) 及 Onset(x) 的地方，計算 Rest 減去其前面的 Onset 之值，若大於零則表示為切分音之處。例如：圖 2.17 中， s_0 之值為 $-2 - (-4) = 2 > 0$ ， s_1 之值為 $-1 - (-3) = 2 > 0$ ， s_2 之值為 $-2 - (-4) = 2 > 0$ ，代表此小節有三個切分音，出現位置分別在 s_0 、 s_1 及 s_2 三處。而整個小節的複雜度大小為 $2(s_0) + 2(s_1) + 2(s_2) = 6$ 。

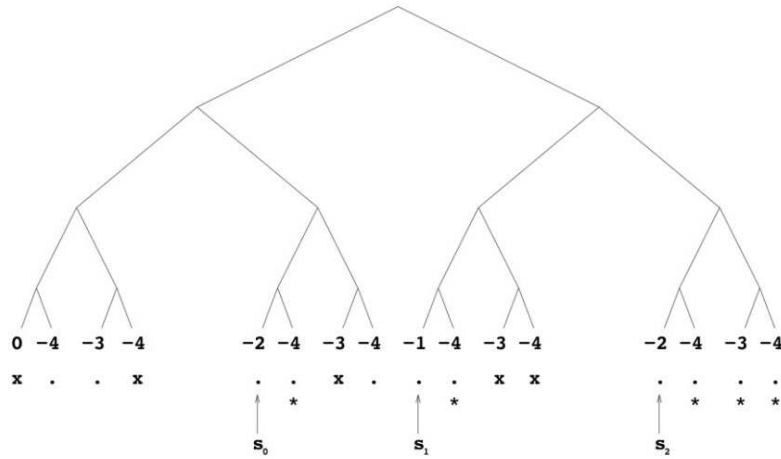


圖 2.17 LHL 複雜度分析例子 [23]

2.6 自動作曲

本論文的主要目的為使用 Pitch-Class Set 為理論基礎來自動作曲，自動作曲是利用電腦產生符合樂理規則的音樂，過去關於自動作曲的研究中，基本上分為兩種：一種是根據樂理或經驗來產生規則，一種是利用機器學習方法，由訓練資料學習而產生出作曲規則。以下介紹自動作曲的相關研究及其技術發展。

2.6.1 Melody Generator [6]

Dirk-Jan Povel 所開發的自動作曲程式---Melody Generator，其利用基本樂理及和聲來產生旋律，但無特定音樂風格。使用者可逐步照著作者設計的 Model 來產生旋律，不同 model 所需的設定也不同，也可選擇全自動產生音樂。另外，此程式還能讓使用者手動編輯產生後的音樂。

圖 2.18 為 Melody Generator 的程式介面，從圖中可看出作者設計了許多參數供使用者調整，整個介面分為七大部分，使用者可自由選擇及調整自己需要的部分。在旋律結構部分，作者提供五種 model，其中，ChordBasedMode 1 可以設定結構，讓產生出的旋律較有完整性；另外還有 AttactionMode 1、ScaleBasedModel、Basic 及 DrumSoloModel。

從圖 2.19 中可了解 Melody Generator 產生旋律的基本步驟，以 ChordBasedModel 為例：首先決定調性及產生節奏，程式將節奏點分為一小節 16 點，依據各點的權重來決定是否產生音符 (Step 1)；接著設定 Gap，Gap 指的是樂句間的時間，圖 2.19 的例子設定的 Gap 為 2，表示此樂句與下一樂句間的休息長度；Step 3 為產生和聲，系統預設為終止式，使用者可以選擇系統內建的終止式，也可自訂；Step 4 設定旋律走向，“+”表示旋律往上行進，“-”表示旋律往下行進；之後系統會根據節奏、和聲及旋律走向，在正拍上填入骨幹音 (Step 5)；最後則加上裝飾音完成樂句 (Step 6)。

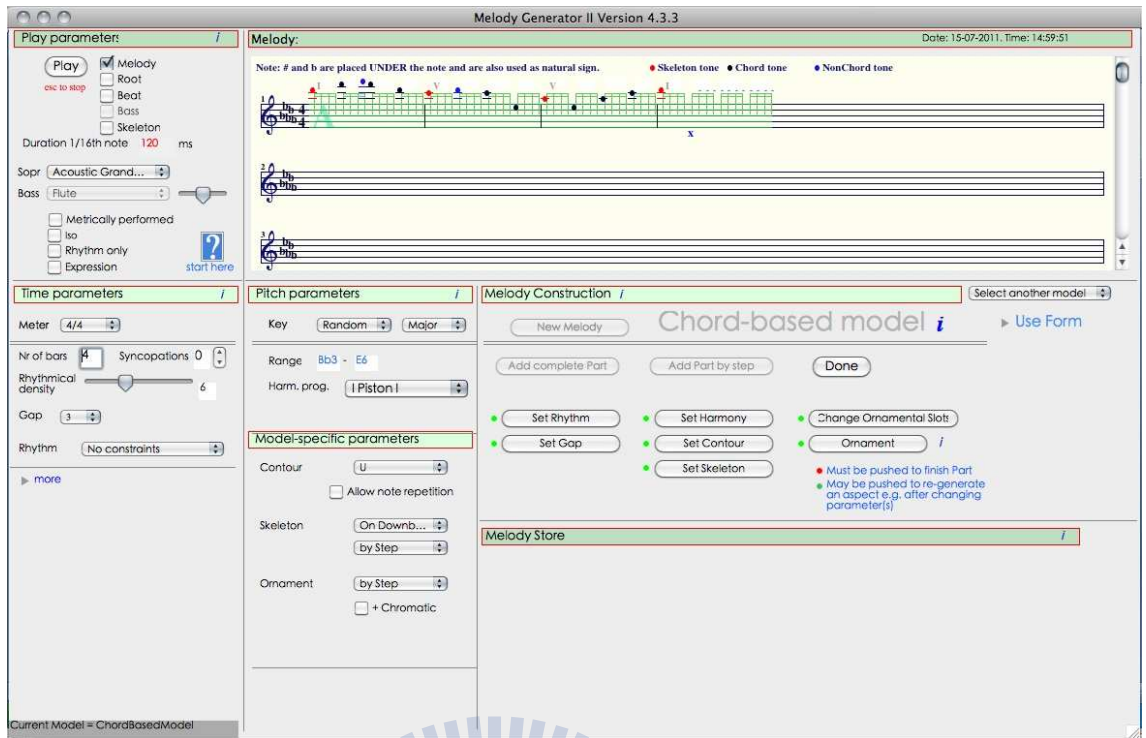
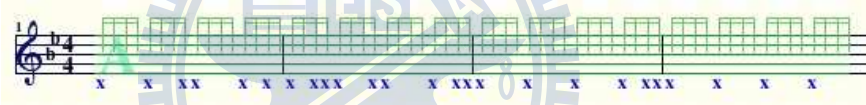


圖 2.18 Melody Generator 程式介面

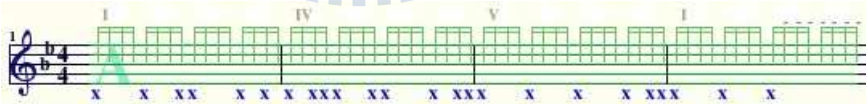
Step 1 : Set Rhythm



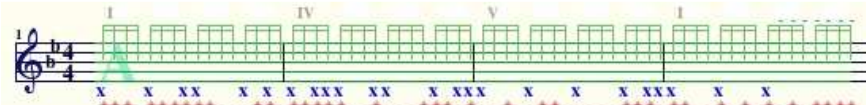
Step 2 : Set Gap



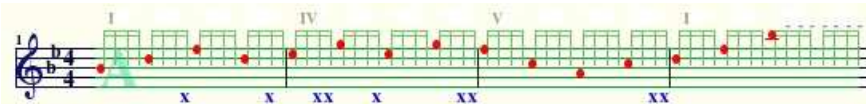
Step 3 : Set Harmony



Step 4 : Set Contour



Step 5 : Set Skeleton



Step 6 : Ornament

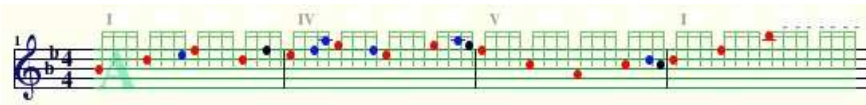


圖 2.19 Melody Generator 產生旋律步驟圖

2.6.2 The Algorithmic Composer: ALICE [24]

ALICE (Algorithmically Integrated Composing Environment) 是由 David Cope 所研發 [3]，它可以幫助作曲者延伸想法、提供作曲者一些音樂片段、甚至完成其未完成的作品。ALICE 藉由分析音樂片段裡的各種參數來建立資料庫，並產生規則，程式會根據這些規則產生音樂，發展出作曲者特有的風格。

ALICE 的特點是使用 Pattern Match 找出 “Signature”、”Earmark” 和 “Unification”。 “Signature” 指的是音樂特徵，”Earmark” 是連結段落間的重要特徵， “Unification” 為音樂當中重複出現的樣式，使音樂具有其獨特性。

另外，ALICE 使用 Pitch Class 來建立和聲規則，這樣可以取得所有的和聲種類，因此也較適合產生無調性的音樂。ALICE 將音樂切成許多區塊，並記錄區塊內使用的 Pitch Class，還有儲存音樂前後文連接的規則，在自動創作過程中，會比較現在產生片段的 Pitch Class 和資料庫中記錄的規則，從中選出符合的規則來產生下個音樂片段。

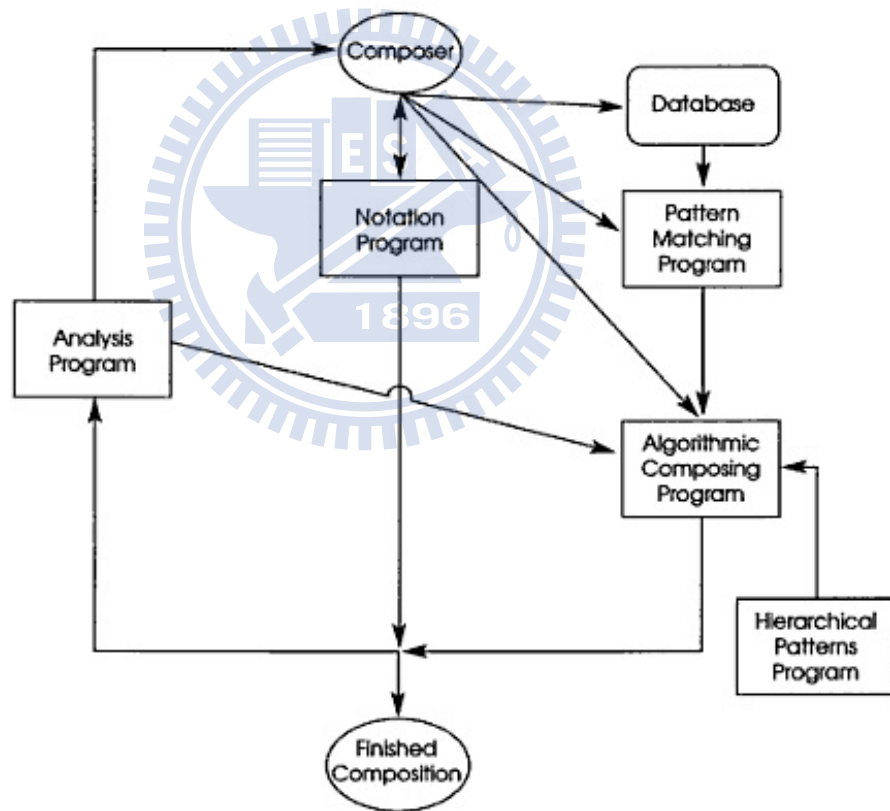


圖 2.20 ALICE 作曲流程圖

資料來源: The Algorithmic Composer [24]

圖 2.20 為 ALICE 的作曲流程，一開始先從之前作曲時所建立的資料庫中取出音樂規則，以及設定不同的程式參數，接著利用這些資料來進行作曲，使用 Pattern Match 等演算法來產生新的音樂。

第三章、研究方法

在文獻回顧中，電腦自動作曲的方法基本上分為兩種，一種是根據樂理或經驗來產生規則，一種是利用機器學習方法，由訓練資料學習而產生出作曲規則。我們採用的是第一種方法---根據樂理來產生作曲規則，使用樂理可以獲得全面性規則，不會偏頗，也具有理論依據，所以用規則去作曲會更貼近音樂的本質 [25]。我們針對調性、曲式及和聲，作為作曲規則來產生旋律。

本論文的主要目的為使用 Pitch-Class Set 為理論基礎來自動作曲，Pitch-Class Set 是一組無序的 Pitch-Class 集合，每組 Pitch-Class Set 則可由一獨特的 Prime Form 表示，若兩組 Set 的 Prime Form 相同，則表示它們包含相同數目的 Pitch 及音程，聽起來的感覺相差不大，因此我們使用 Prime Form 及其 Transposition、Inversion、 T_nI 的變化來當作音高產生的主要依據。

3.1 系統架構

圖 3.1 為本論文的系統流程圖，步驟說明如下：

第一個步驟是找出 Prime Form。系統提供兩種輸入方式：一種為輸入一個 MIDI File，一種則是讓使用者自由選擇 Pitch，系統會根據輸入的資料找出 Prime Form，來當作音高產生的依據。

接著系統會根據使用者選擇的 Complexity 的值來產生節奏，系統以小節為單位，隨機選擇可用的組合；我們使用 LHL 反推法來產生節奏，產生原理及節奏組合於 3.3 節詳細介紹。

步驟三我們提供非調性及調性的選項讓使用者選擇，由於 Pitch-Class Set 本來是分析 12 音列的方法，12 音列為非調性音樂，因此利用 Prime Form 變化產生非調性的旋律是合理的，且非調性音樂並無限制，系統也非產生 12 音列的音樂，因此並不需要 Prime Form Filter。而在調性音樂的部分，由於調性音樂需符合調性音階，因此需使用 Prime Form Filter 來篩選出符合的 Prime Form 變化。

第四步驟則將第二步驟產生的節奏組合與第三步驟 Prime Form 變化產生的音高來產生旋律。為了讓調性音樂聽起來有方向性，系統會搭配終止式讓產生的旋律有和聲架構；產生旋律的同時也考慮曲式的架構，因此我們加入動機及主題變化的概念，讓產生旋律更有結構性，系統依照使用者選擇的小節長度來建立架構，前二小節為動機，八小節為主題，17 小節以上才會有主題變化。

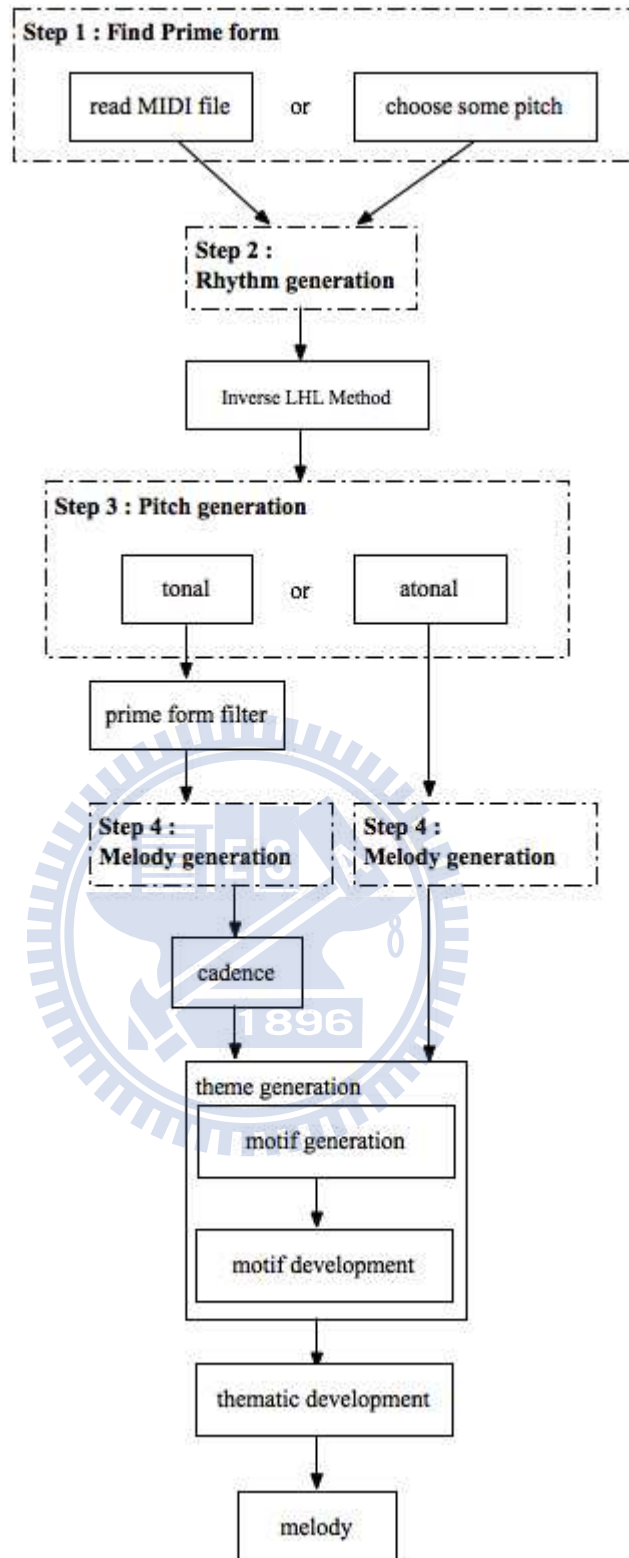


圖 3.1 系統流程圖

3.2 找出 MIDI 音樂的 Prime Form

從 MIDI File 裡找出 Prime Form 之前，必須先分析 MIDI file，取出需要的資料後再找出 Prime Form。Prime Form 是 Pitch-Class Set 裡最 Normal 的 Normal Form，因此我們分析 MIDI File 後，會先找出此 MIDI File 的所有 Pitch，接著將這些 Pitch 轉成 Pitch Class，再從這些 Pitch Class 中找出 Normal Form。



圖 3.2 「櫻花」譜例

以圖 3.2 為例，歌曲「櫻花」整首曲子出現的 Pitch Class 有：2,4,5,9,10，其 Normal Form 為 [0,2,3,7,8]，Inversion Normal Form 為 [0,1,5,6,8]，比較結果是 [0,1,5,6,8] 較 Packed，因此 [0,1,5,6,8] 為「櫻花」這首曲子的 Prime Form。

3.3 節奏產生方法

系統產生節奏的方法分為兩種，為 3/4 拍及 4/4 拍，分別產生三拍及四拍的音樂。3.3.1 節說明 3/4 拍的產生方法，3.3.2 節為 4/4 拍的產生方法。

3.3.1 3/4 拍節奏產生方法

3/4 拍的節奏是根據常見 3/4 拍節奏加入系統內建資料庫，當使用者選擇 3/4 拍時，系統會從這些節奏樣本中，以一小節為單位，隨機選出其中一種來產生該小節的節奏。

表 3.1 為系統內建的所有節奏樣本，數值 1.0 代表四分音符，數值 0.5 代表八分音符，數值 0.25 代表十六分音符，數值 1.5 為附點四分音符，數值 2.0 代表二分音符，數值若為負號則為休止符。

表 3.1 系統內建的 3/4 拍節奏樣本資料庫

| | |
|-----|--|
| 1. | {1.0, 1.0, 1.0} |
| 2. | {1.0, 0.5, 0.5, 1.0} |
| 3. | {0.5, 0.5, 0.5, 0.5, 0.5, 0.5} |
| 4. | {0.5, 0.5, 0.5, 0.5, 1.0} |
| 5. | {0.5, 0.5, 1.0, 1.0} |
| 6. | {1.0, 0.25, 0.25, 0.25, 0.25, 0.5, 0.5} |
| 7. | {1.5, 0.5, 1.0} |
| 8. | {0.5, 0.5, 2.0} |
| 9. | {1.0, 0.75, 0.25, 1.0} |
| 10. | {1.0, 0.5, 0.5, 0.5, 0.5} |
| 11. | {1.0, 2.0} |
| 12. | {0.5, 0.5, 2.0} |
| 13. | {0.5, 1.0, 0.5, 1.0} |
| 14. | {0.5, 2.5} |
| 15. | {2.5, 0.5} |
| 16. | {-0.5, 0.5, 2.0} |
| 17. | {-0.5, 0.5, 0.5, 0.5, 1.0} |
| 18. | {-1.0, 0.5, 0.5, 1.0} |
| 19. | {-0.25, 0.25, 0.25, 0.25, 0.5, 0.5, 1.0} |

3.3.2 4/4 拍節奏產生方式

由於複雜度是節奏的特徵之一，而音樂節奏的複雜度可用節奏複雜度分析方法 [17, 18] 來定義，其中又以 LHL 分析法效果最好，因此本論文在 4/4 拍的節奏產生方法，是以 LHL 節奏複雜度分析法的反推來實作。

● LHL 反推法

LHL 在分析節奏複雜度的方法裡效果最好，因此我們嘗試採用其反推法來產生節奏片段。以下說明 LHL 反推法的過程，圖 3.3 為 LHL 的樹狀圖，上面標示 LHL 的階層及權重值。

圖 3.3 中，樹狀圖的權重“-1”、“-2”、“-3”、“-4”分別代表 2 分音符、4 分音符、8 分音符、16 分音符；“x”代表 Onset 的位置，“.”代表非 Onset 的位置，可能為休止符或是 x 位置的延音；接著我們用英文字母來代表有複雜度出現的地方，即 a、b、c、d、e、f、g 這七個位置，也就是“.”的權重值減去“x”權重值大於零之處，而每個位置的權重值，計算方法是把“.”的權重減去前面“x”的權重，例如在位置 a，其權重值為 $-3 - (-4) = 1 > 0$ 。

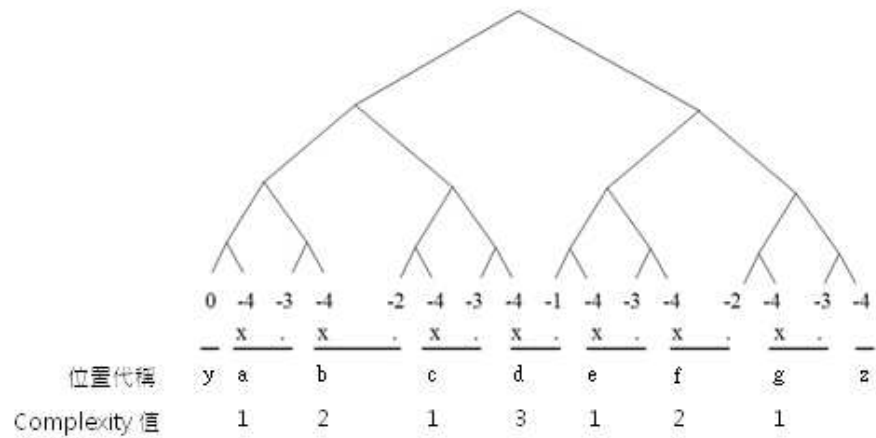


圖 3.3 LHL 反推法之樹狀圖(4/4, 16-baet)

表 3.2 複雜度組合

| 欲產生之 Complexity 值 | | 產生方法：(n) 代表選 n 個 |
|-------------------|---------------|---------------------------|
| 1 | 1 | a.c.e.g (1) |
| 2 | 1+1 | a.c.e.g (2) |
| | 2 | b.f(1) |
| 3 | 1+1+1 | a.c.e.g (3) |
| | 1+2 | a.c.e.g (1) + b.f (1) |
| | 3 | d |
| 4 | 1+1+1+1 | a.c.e.g (4) |
| | 1+1+2 | a.c.e.g (2) + b.f (1) |
| | 2+2 | b.f (2) |
| | 1+3 | a.c.e.g (1) + d |
| 5 | 1+1+1+2 | a.c.e.g (3) + b.f (1) |
| | 1+1+3 | a.c.e.g (2) + d |
| | 1+2+2 | a.c.e.g (1) + b.f (2) |
| | 2+3 | b.f (1) + d |
| 6 | 1+1+1+1+2 | a.c.e.g (4) + b.f (1) |
| | 1+1+1+3 | a.c.e.g (3) + d |
| | 1+1+2+2 | a.c.e.g (2) + b.f (2) |
| | 1+2+3 | a.c.e.g (1) + b.f (1) + d |
| 7 | 1+1+1+1+3 | a.c.e.g (4) + d |
| | 1+1+1+2+2 | a.c.e.g (3) + b.f (2) |
| | 1+1+2+3 | a.c.e.g (2) + b.f (1) + d |
| | 2+2+3 | b.f (2) + d |
| 8 | 1+1+1+1+2+2 | a.c.e.g (4) + b.f (2) |
| | 1+1+1+2+3 | a.c.e.g (3) + b.f (1) + d |
| | 1+2+2+3 | a.c.e.g (1) + b.f (2) + d |
| 9 | 1+1+1+1+2+3 | a.c.e.g (4) + b.f (1) + d |
| | 1+1+2+2+3 | a.c.e.g (2) + b.f (2) + d |
| 10 | 1+1+1+2+2+3 | a.c.e.g (3) + b.f (2) + d |
| 11 | 1+1+1+1+2+2+3 | a.c.e.g (4) + b.f (2) + d |

定義出七個位置的權重值後，接著計算 Complexity 的組合。不同 Complexity 的值有不同種的組合，例如：Complexity 為 3 時，組合有三種，若組合是 1+2，則表示從權重 1 的 a、c、e、g 四個位置中選一個，再從權重 2 的 b、f 兩個位子中選一個，最後此小節裡會有兩個 Note。表 3.2 列出所有 Complexity 組合的種類。

然而節奏的組合並不只有表 3.2 的內容，我們透過加音但不影響節奏複雜度的方法，來增加可用的節奏組合，圖 3.3 裡的“y”及“z”的位置為一個小節內第一個 Beat 及最後一個 Beat，頭尾的位置不論是否有 Onset 皆不影響節奏複雜度的值，因此頭尾的部分可隨機選擇加或不加，組合有三種，如表 3.3。

表 3.3 小節頭尾加音組合表

| 位置 組合 | y | z |
|----------|-------|-------|
| 組合 1 | onset | |
| 組合 2 | | onset |
| 組合 3 | onset | onset |
| 組合 4 | | |

接著是在有權重的位置隨機加音，此步驟的狀況會有兩種：若隨機位置在 Onset 的地方，則和它一組的“.”也會被選到，如此才不會增加其複雜度；若隨機位置在“.”的地方，則檢查和它一組的“x”是否為 Onset，如果和它一組的“x”是 Onset，表示此位置不可為 Onset，否則會減少複雜度，若和它一組的“x”沒有 Onset，則可加 Onset 在此位置上。

3.4 音高產生

找出 Prime Form 後，系統會先找出其所有變化，即 Origin、Inversion、 $T_n(n=1\sim 11)$ 及 $T_nI(n=1\sim 11)$ 。 T_n 是將 Prime Form 內的所有數字作加 n 的運算，若運算後的數字大於 12，則須取 mod 12 後的數字，例如： $[11,0,4] = T_{11}[0,1,5]$ ；Inversion 是用 12 去減掉 Prime Form 內的所有數字，例如： $[0,7,11] = T_0I[0,1,5]$ ； T_nI 為綜合 Transposition 及 Inversion 的變化，例如： $[1,2,9] = T_2I[0,1,5]$ 。

接著系統會依照非調性或調性作 Prime Form 變化的篩選：在非調性時，因為沒有音階限制，所以 Prime Form 的全部變化皆可使用；在調性時，需符合大調音階或小調音階，在 3.4.3 節裡，我們將會介紹 Prime Form 在調性時的篩選方法。表 3.3 為 Prime Form 的所有變化對照表。篩選出可用的 Prime Form 後，系統會以這些 Pitch-Class set 為依據，並加上 Pitch Offset 來產生音高。

表 3.4 Prime form 所有變化對照表 (以 [0,3,7] 為例)

| 變化 | Prime form 變化 | |
|---|---------------|----------|
| Origin | [0,3,7] | |
| Inversion | [0,9,5] | |
| Transposition ($T_n, n = 1\sim 11$) | T_1 | [1,4,8] |
| | T_2 | [2,5,9] |
| | T_3 | [3,6,10] |
| | T_4 | [4,7,11] |
| | T_5 | [5,8,0] |
| | T_6 | [6,9,1] |
| | T_7 | [7,10,2] |
| | T_8 | [8,11,3] |
| | T_9 | [9,0,4] |
| | T_{10} | [10,1,5] |
| | T_{11} | [11,2,6] |
| Transposition+Inversion ($T_nI, n = 1\sim 11$) | T_1I | [1,10,6] |
| | T_2I | [2,11,7] |
| | T_3I | [3,0,8] |
| | T_4I | [4,1,9] |
| | T_5I | [5,2,10] |
| | T_6I | [6,3,11] |
| | T_7I | [7,4,0] |
| | T_8I | [8,5,1] |
| | T_9I | [9,6,2] |
| | $T_{10}I$ | [10,7,3] |
| | $T_{11}I$ | [11,8,4] |

3.4.1 非調性

非調性音樂的限制少，因此 Prime Form 的所有變化皆可使用。系統以每一小節為單位，每小節從 Prime Form 的所有變化中隨機選擇一組，再搭配隨機的 Pitch Offset 來產生音高。例如：此小節的 Prime Form 變化是 [2,6,9]，Pitch Offset 是 53，則此小節的組成音為 55、59 及 62。

3.4.2 調性

調性音樂分為大小調共 24 調，小調音階又分為旋律小音階及和聲小音階，系統採用的是和聲小音階，表 3.4 為 24 大小調音階與 Pitch Class 對照表。

表 3.5 大小調音階和 Pitch Class 對照表

| 大調音階 | | | | | | | |
|---------------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| C 大調音階 | C | D | E | F | G | A | B |
| Pitch class | 0 | 2 | 4 | 5 | 7 | 9 | 11 |
| D ^b 大調音階 | D ^b | E ^b | F | G ^b | A ^b | B ^b | C |
| Pitch class | 1 | 3 | 5 | 6 | 8 | 10 | 0 |
| D 大調音階 | D | E | F [#] | G | A | B | C [#] |
| Pitch class | 2 | 4 | 6 | 7 | 9 | 11 | 1 |
| E ^b 大調音階 | E ^b | F | G | A ^b | B ^b | C | D |
| Pitch class | 3 | 5 | 7 | 8 | 10 | 0 | 2 |
| E 大調音階 | E | F [#] | G [#] | A | B | C [#] | D [#] |
| Pitch class | 4 | 6 | 8 | 9 | 11 | 1 | 3 |
| F 大調音階 | F | G | A | B ^b | C | D | E |
| Pitch class | 5 | 7 | 9 | 10 | 0 | 2 | 4 |
| G 大調音階 | G | A | B | C | D | E | F [#] |
| Pitch class | 7 | 9 | 11 | 0 | 2 | 4 | 6 |
| A ^b 大調音階 | A ^b | B ^b | C | D ^b | E ^b | F | G |
| Pitch class | 8 | 10 | 0 | 1 | 3 | 5 | 7 |
| A 大調音階 | A | B | C [#] | D | E | F [#] | G [#] |
| Pitch class | 9 | 11 | 1 | 2 | 4 | 6 | 8 |
| B ^b 大調音階 | B ^b | C | D | E ^b | F | G | A |
| Pitch class | 10 | 0 | 2 | 3 | 5 | 7 | 9 |
| B 大調音階 | B | C [#] | D [#] | E | F [#] | G [#] | A [#] |
| Pitch class | 11 | 1 | 3 | 4 | 6 | 8 | 10 |

| 小調音階 | | | | | | | |
|---------------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| c 小調音階 | c | d | e ^b | f | g | a ^b | B |
| Pitch class | 0 | 2 | 3 | 5 | 7 | 8 | 11 |
| c [#] 小調音階 | c [#] | d [#] | e | f [#] | g [#] | a | c |
| Pitch class | 1 | 3 | 4 | 6 | 8 | 9 | 0 |
| d 小調音階 | d | e | f | g | a | b ^b | c [#] |
| Pitch class | 2 | 4 | 5 | 7 | 9 | 10 | 1 |
| e ^b 小調音階 | e ^b | f | g ^b | a ^b | b ^b | b | d |
| Pitch class | 3 | 5 | 6 | 8 | 10 | 11 | 2 |
| e 小調音階 | e | f [#] | g | a | b | c | d [#] |
| Pitch class | 4 | 6 | 7 | 9 | 11 | 0 | 3 |
| f 小調音階 | f | g | a ^b | b ^b | c | c [#] | e |
| Pitch class | 5 | 7 | 8 | 10 | 0 | 1 | 4 |
| f [#] 小調音階 | f [#] | g [#] | A | b | c [#] | d [#] | f |
| Pitch class | 6 | 8 | 9 | 11 | 1 | 3 | 5 |
| g 小調音階 | g | a | b ^b | c | d | e ^b | f [#] |
| Pitch class | 7 | 9 | 10 | 0 | 2 | 3 | 6 |
| g [#] 小調音階 | g [#] | a [#] | b | c [#] | d [#] | e | g |
| Pitch class | 8 | 10 | 11 | 1 | 3 | 4 | 7 |
| a 小調音階 | a | B | c | d | e | f | g [#] |
| Pitch class | 9 | 11 | 0 | 2 | 4 | 5 | 8 |
| b ^b 小調音階 | b ^b | C | d ^b | e ^b | f | g ^b | a |
| Pitch class | 10 | 0 | 1 | 3 | 5 | 6 | 9 |
| b 小調音階 | b | c [#] | d | e | f [#] | g | a [#] |
| Pitch class | 11 | 1 | 2 | 4 | 6 | 7 | 10 |

3.4.3 Prime form filter

由於調性音樂的組成音必須在符合的音階內，因此需要設計 Filter 來篩選出符合條件的 Prime Form 變化，使用者選擇大調時，系統會根據大調音階來選出合

適的 Prime Form 變化；而使用者選擇小調時，系統則根據和聲小音階的組成來選出可用的 Prime Form 變化。

圖 3.4 為 Prime Form 在調性時的 Filter 示意圖，以 Prime Form [0,3,7] 為例，在 C 大調時，從原始 Prime Form 及其變化中，篩選出來可用的有：[0,9,5] (Inversion)、[2,5,9] (T₂)、[4,7,11] (T₄)、[9,0,4] (T₉)、[2,11,7] (T₂I)、[7,4,0] (T₇I)；在 c 小調時，篩選出可用的則有：[0,3,7] (Oring)、[5,8,0] (T₅)、[8,11,3] (T₈)、[2,11,7] (T₂I)、[3,0,8] (T₃I)、[10,7,3] (T₁₀I)。

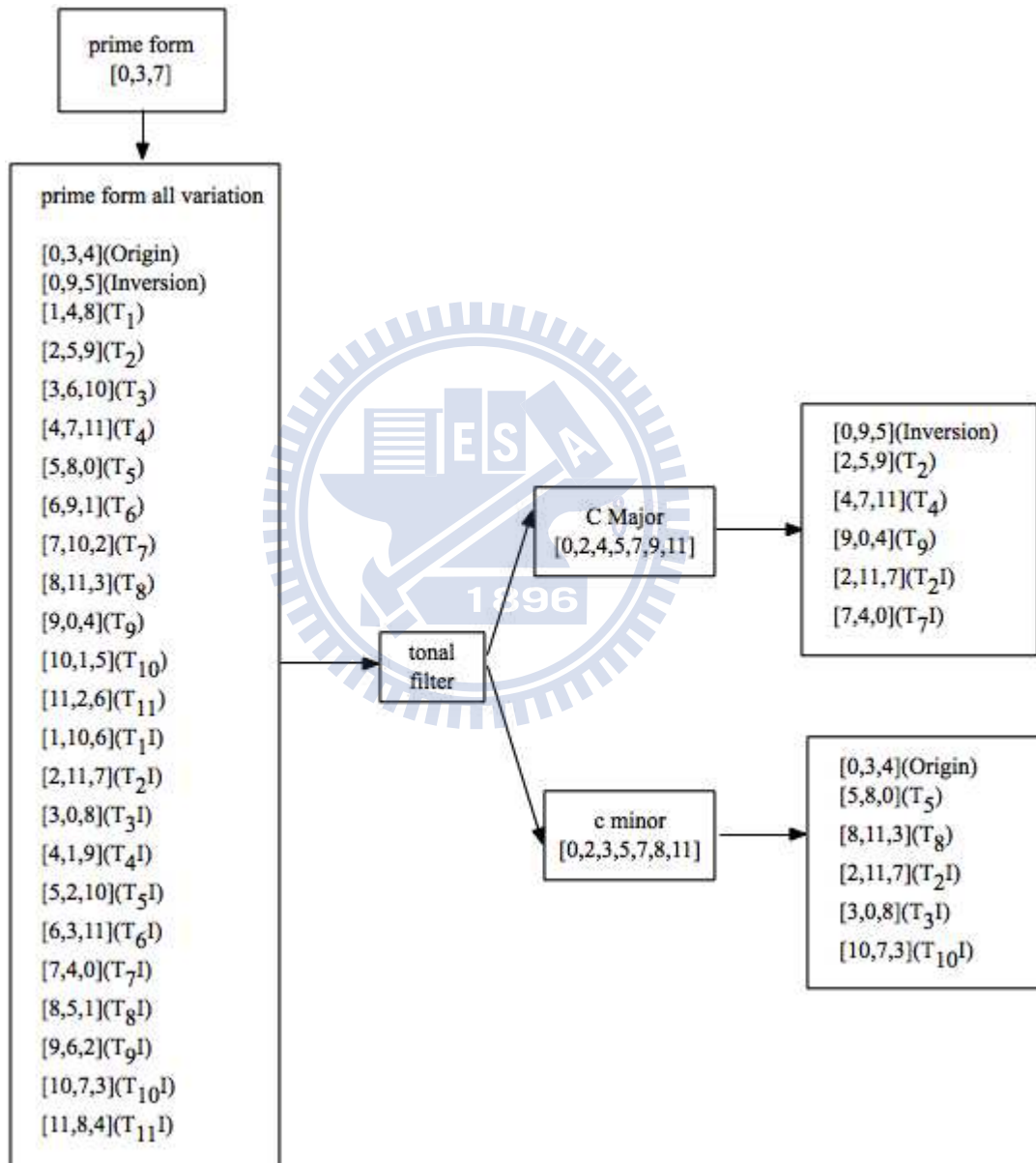


圖 3.4 Prime form tonal filter 示意圖

3.4.4 終止式 (Cadence)

終止式是樂曲中樂段或樂句停頓處,可視為音樂的暫時停頓或結束。為了讓產生的調性音樂更聽得出調性,我們加入終止式的設定來增加其和聲結構性。終止式分為正格終止 (Authentic Cadence)、變格終止 (Plagal Cadence)、假終止 (Deceptive Cadence) 及半終止 (Half Cadence) 等,最常見的為正格終止及變格終止,其最後一個和絃必須是主和絃 (I), 倒數第二個和絃又依屬和絃 (V) 及下屬和絃 (IV) 各分別屬於正格終止 (圖3.5) 及變格終止 (圖3.6)。



圖3.5 正格終止 (Authentic cadence)

資料來源：http://en.wikipedia.org/wiki/File:IV-V-I_in_C.png



圖3.6 變格終止 (Plagal cadence)

資料來源：http://en.wikipedia.org/wiki/File:Plagal_cadence_in_C.png

然而簡單的樂曲大部分使用的是正格終止,通常與正格終止搭配的和聲進行為 $I \rightarrow IV \rightarrow V \rightarrow I$ (圖3.7) 及 $I \rightarrow II^6 \rightarrow V \rightarrow I$ (圖3.8), 因此系統中以四小節為一個終止式單位,套用其中一組當作和聲進行。



I IV V I

圖3.7 終止式 $I \rightarrow IV \rightarrow V \rightarrow I$



圖3.8 終止式 I → II⁶ → V → I

雖然篩選出的 Prime Form 變化皆在音階內音上，但搭配和聲時，若出現和絃外音，聽起來會不和諧，因此在搭配和聲時，系統預設一半的機率為和絃內音，另一半的機率為 Prime Form 篩選出的音高，其中 Prime Form 篩選出的音高裡有一定機率包含和絃內音。

3.5 動機及主題發展

為了讓產生的旋律聽起來較有結構性，我們加入動機及主題的概念。產生節奏及音高後，接著系統會產生動機及主題發展變化。

3.5.1 動機

動機的長度通常為兩小節，由一組特有的音程及節奏構成，圖 3.9 為主題及動機的基本架構，圖中前兩小節為動機。所以本系統設定第一及第二小節為旋律產生器的動機，其節奏根據使用者所選的複雜度而產生，音高則由 Prime Form 的變化為基本 Pitch Class 組成，再加上 Pitch Offset 而成。

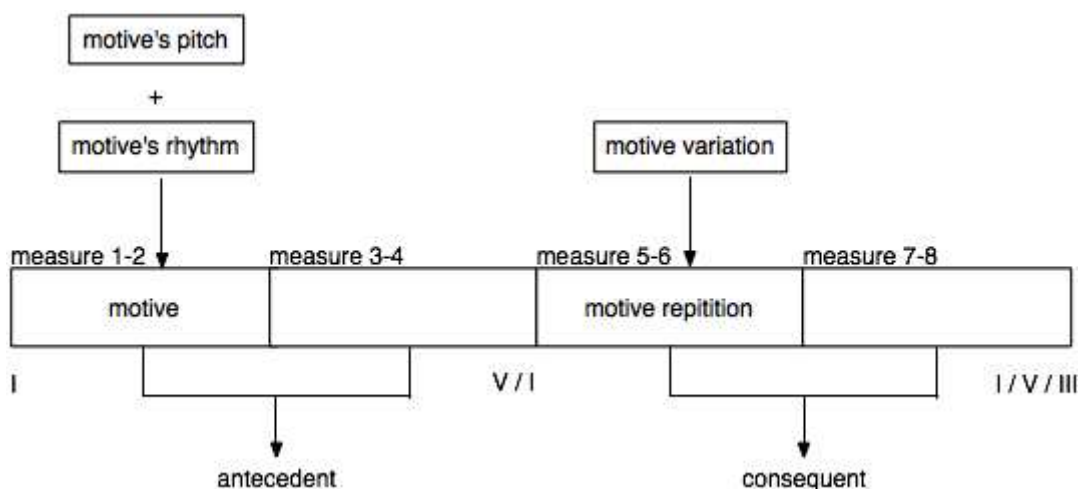


圖 3.9 主題與動機

3.5.2 主題

主題由動機變化而產生，通常是一完整的樂句或片段 [10]。本論文中，我們定義主題的長度是八小節。第一至第四小節稱為“antecedent”，第五至第八小節則稱為“consequent”，“consequent”的特性為“antecedent”的重複變化。第一及第二小節是動機，長度是兩小節，其重複與變化會出現在第五及第六小節。“antecedent”通常結束在 V 級，有時候在 I 級；“consequent”則通常結束在 I、V 或 III 級，為完全終止式。由於系統採用正格終止，因此系統裡的“antecedent”及“consequent”皆結束在 I 級。圖 3.9 為動機發展成一主題的架構範例，第三、四小節及第七、八小節是沒有動機的，為獨立的音樂片段。

依照 3.4.4 節所提到，系統採用的終止式為正格終止，因此在第四及第八小節時，皆結束在 I 級，所以在第四小節並不會有結束在 V 級的狀況，第八小節也不會有結束 III 級及 V 級的狀況。圖 3.10 為本論文的系統主題架構圖，小節下方標示出該小節有可能搭配的和絃。

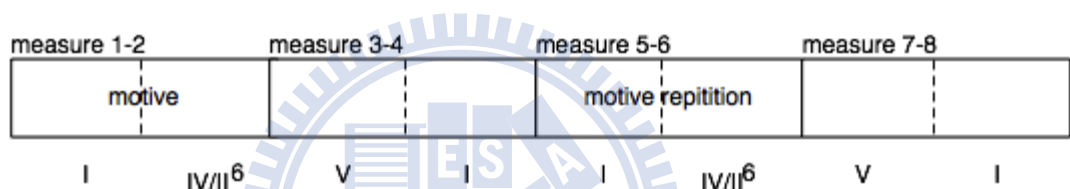


圖 3.10 系統主題架構

3.5.3 動機變化

相關研究裡我們介紹動機的重複變化，分別是“Exact”、“Modified”、及“Developed”。 “Exact”方法保留所有特徵及關係，“Modified”方法則會改變原有動機的一些特徵及保留剩餘的部分。

“Exact”方法保留所有特徵及關係，聽者較容易辨識出動機變化的不同，因此系統使用“Exact”的方法來作動機重複變化，“Exact”方法包括 Transposition、Inversion、Retrograde、Retrograde Inversion、Diminutions 及 Augmentation，其中，Diminutions 及 Augmentation 會改變整體的結構，所以我們不使用這兩種方法來作動機重複變化。根據圖 3.9，動機變化會出現在八小節主題的第五及第六小節。

由於旋律產生分為調性及非調性，因此在調性的情況下，所有動機變化必須符合調性音階，在 Transposition 及 Inversion 時，系統會判斷產生的 Pitch 是否符合 Tonal Transposition 和 Tonal Inversion；Retrograde 因為 Pitch 變化和原本一樣，不需要判斷變化是否符合調性音階。

3.5.4 主題發展

主題發展的方式包括 Transposition 、 Interval Expansion and Contraction 、 Rhythmic Augmentation and Diminution 、 Inversion 、 Tonality Changes 等，其中 Transposition 及 Inversion 的方法較容易聽出來發展性，因此系統選用此兩種作法來作主題變化，以圖 3.11 為例，(a) 是原本的主題，(b) 為作 Transposition 變化後的主題，(c) 則是 Inversion 的變化。

一首簡單的旋律型式可由 A 、 B 及其變化組成，因此我們以三段式 (ABA') 為例，在曲式 ABA' 時，主題 A 的發展出會現在第 17 至第 24 小節，使用者若選擇 16 小節以上的選項時，系統才會產生有主題發展的旋律。



圖 3.11 系統採用的主題發展方法

第四章、系統功能與實作

本章介紹系統功能與實作的結果，我們設計一個使用者圖形介面供使用者方便操作，系統提供許多控制選項引導使用者使用，並能播放產生的旋律及印出樂譜。4.1 節我們將逐項介紹系統細部的功能，4.2 節則討論實作的結果。

4.1 系統功能

圖 4.1 為系統介面圖，其中包括四大步驟：第一步為是否讀入 MIDI File，若使用者不希望讀入 MIDI File，在第二步時會讓使用者自由選用想創作的 Pitch，系統會依據這些 Pitch 找出他們所屬的 Prime Form；若使用者選擇讀入 MIDI File，到第二步時則會顯示 MIDI File 的 Prime Form。接著分為產生 Tonal 音樂或 Atonal 音樂，選擇產生 Tonal 音樂後，會有大小調及旋律 Contour 的選項出現；選擇 Atonal 則直接進入第四步驟。第四步驟時，系統提供 Meter、Bar Number 及 Rhythm Complexity 三個選項，最後則是根據前四個步驟的選擇產生旋律，並可選擇播放、停止及印出樂譜。



圖 4.1 系統介面圖

讀入 MIDI File 後，系統會統計此 MIDI File 中，各個 Pitch Class 出現的比率，也就是組成音有哪些，並找出它的 Prime Form，顯示於 STEP 2 的方框中，如圖 4.2 所示。

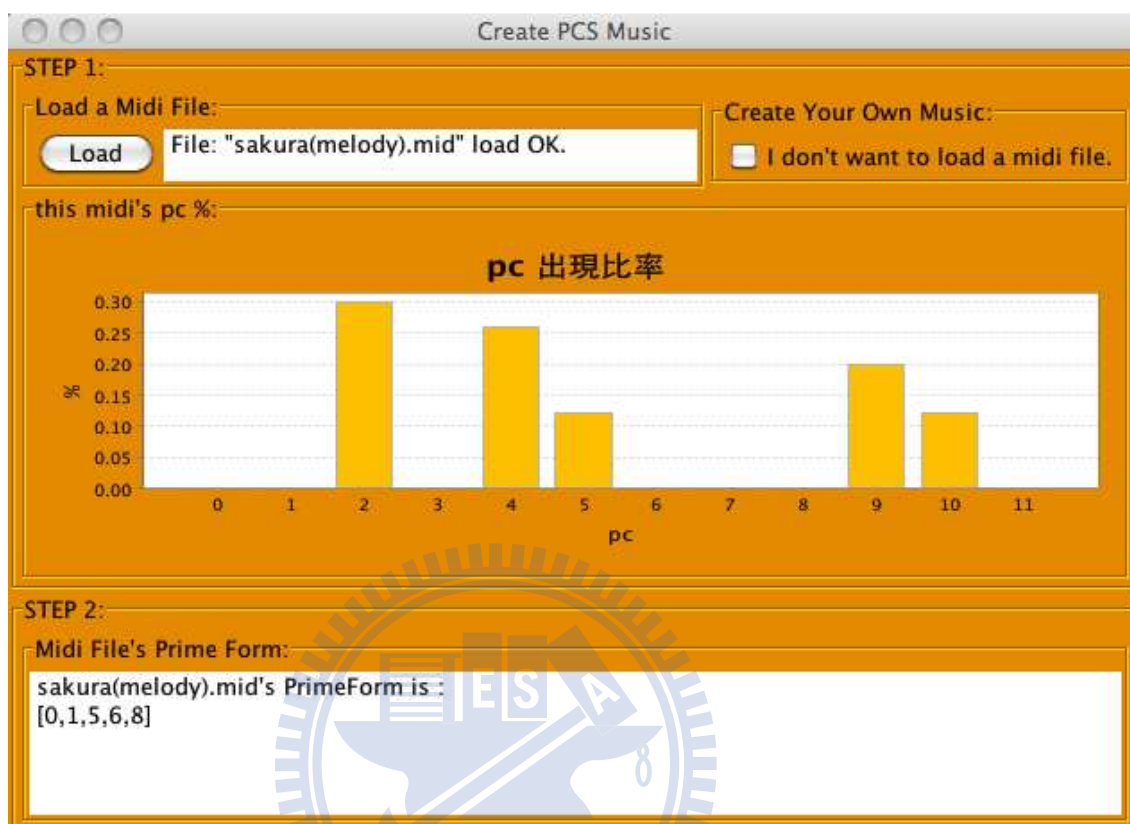


圖 4.2 讀入 MIDI File

若使用者選擇不讀入 MIDI File，在 STEP 2 時會顯示 C 至 B 等 12 個 Pitch 讓使用者勾選欲使用的 Pitch，系統同樣會為這些 Pitch 找出 Prime Form，並顯示於下方的方框中，如圖 4.3 所示。

The screenshot shows the 'Create PCS Music' application window. In STEP 1, the 'Load a Midi File' section has an empty text box and a 'Load' button. The 'Create Your Own Music' section has a checked checkbox labeled 'I don't want to load a midi file.'. In STEP 2, the 'Choose Some Notes' section displays 12 pitch classes with checkboxes: C (checked), C# (unchecked), D (unchecked), D# (unchecked), E (checked), F (unchecked), F# (unchecked), G (checked), G# (unchecked), A (unchecked), A# (unchecked), B (unchecked). Below the checkboxes, a text box contains 'C,E,G, are selected.' and 'Prime form is:[0,3,7]'. A 'Go' button is located to the right of the text box.

圖 4.3 Prime Form 選擇

接著是選擇產生 Tonal 音樂或 Atonal 音樂，若使用者選擇 Tonal 音樂，會有大小調及旋律 Contour 的選項出現，Contour 根據 Melody Generator [23] 的作法，系統提供五種不同的選項，包括四種變化及從中隨機選出一種變化（圖 4.4）。

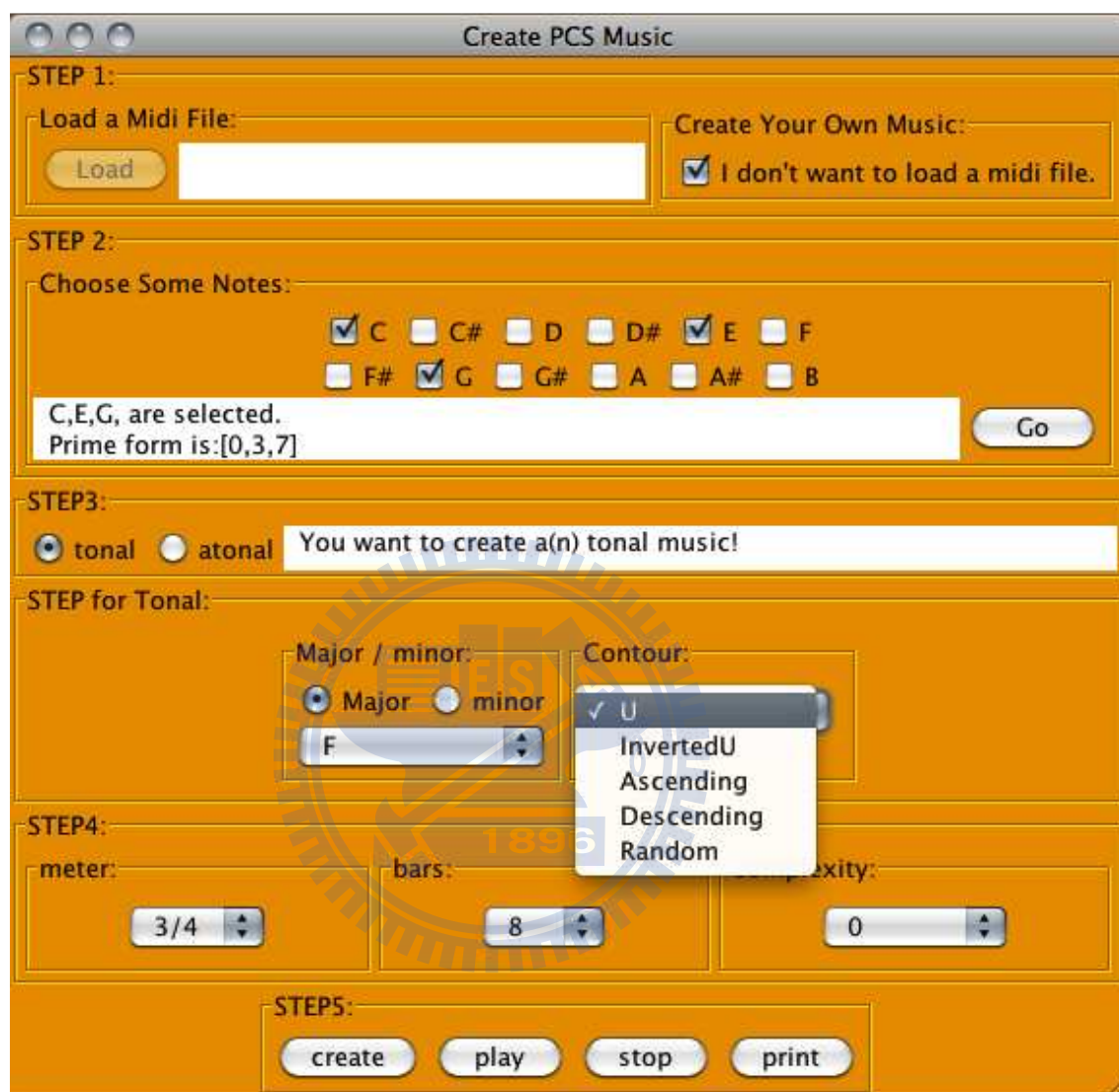


圖 4.4 Tonal 音樂選擇

以下分別介紹 STEP 4 中的三個選項：

- **拍號 (Meter)**

系統提供 3/4 拍及 4/4 拍的選項，分別產生三拍及四拍的音樂。3/4 拍的節奏是根據常見 3/4 拍節奏加入系統內建資料庫，而 4/4 拍的節奏是根據 LHL 反推法來產生。

- **小節數 (Bars)**

系統提供 8、16、24 及 32 小節四種小節數讓使用者選擇，若使用者選擇的小節數為 8 小節，產生的旋律會包括動機變化；若使用者選擇的小節為 16 小節，則會產生包括 A 及 B 兩個主題的旋律；如果選擇的小節為 24 小節以上，產生的旋律會有 A 段的主題發展。

- **節奏複雜度 (Rhythm Complexity)**

我們以 LHL 反推法來產生節奏組合，使用者選擇 Complexity 的數值後，系統便會根據該數值，從表 3.1 中隨機選擇適合該數值的節奏組合，接著利用加音的方法 (3.3 節) 來完成每小節的節奏。

最後，系統根據 STEP 1 至 STEP 4 的選項內容來產生旋律，並依照小節數不同而有不同架構，例如：八小節為一主題並有動機變化，16 小節有 A 及 B 兩種主題，24 小節的形式為 ABA' 等。

4.2 實作結果

我們實作的系統提供 Tonal 及 Atonal 的旋律產生方式，接下來分別介紹 Tonal 及 Atonal 的實作結果。

4.2.1 Atonal 實作結果

圖 4.5 為 Atonal 實作結果的譜例，我們選用 C[#]、D、D[#]、F 當作音高素材，節奏複雜度為 2。系統會先找出 Pitch-Class C[#]、D、D[#]、F 的 Prime Form，也就是 [0,1,2,4]，接著再找出它的所有變化。從圖中可看出，第八小節使用 T₅([5,6,7,9])，用藍色標示於圖 4.5 中。

接著是動機變化及主題的部分：紅色的部分為動機變化，此實作結果的動機變化皆使用到 Retrograde，主題用綠色標示，主題 A，從第一小節至第八小節，主題 B 則從第九小節至第 16 小節。

4.2.2 Tonal 實作結果

圖 4.6 為 Tonal 實作結果的譜例，我們選用 C、E、G 當作音高素材，節奏複雜度為 7，調性為 G 大調，旋律 Contour 為 U 形。系統找出 C、E、

G 的 Prime Form 為 [0,3,7]，經過 Prime Form Filter 篩選出在 G 大調時可用的 Prime Form 變化有： T_4 ([4,7,11])、 T_9 ([9,0,4])、 T_{11} ([11,2,6])、 T_2I ([2,11,7])、 T_4I ([7,4,0]) 及 T_9I ([9,6,2])，第一小節使用 T_2I ([2,11,7])、第七小節則使用 T_9I ([9,6,2])，以藍色標示於圖 4.6 中。

動機變化的部分用紅色標示，此範例的動機變化為 Retrograde。而在結構上，整首音樂只有八小節，因此只有主題 A，以綠色標示。

theme A

Retrograde

$[5,6,7,9] = T_2[0,1,2,4]$

Retrograde

theme B

圖 4.5 Atonal 實作結果

theme A

$[2,11,7] = T_2I[0,3,7]$

Retrograde

$[2,6,9] = T_9I[0,3,7]$

圖 4.6 Tonal 實作結果

第五章、實驗

在實驗方面，我們從三個面向作實驗效果的評估：一、調性辨識率；二、動機變化是否明顯；三、產生的音樂是否好聽。由於本系統提供調性及非調性的選項來產生音樂，若能分辨出調性音樂，則表示我們的研究方法是有用的。而在動機變化方面，如果能讓受試者聽出明顯變化，也顯示在結構上的成功。實驗的受試者為十四位，其中七位為未受過音樂訓練，七位有受過音樂訓練，而這七位受過音樂訓練者之中，學習音樂達十年以上者有四人，五年以上有一人，三至五年有二人，顯示的比例如圖 5.1 所示。

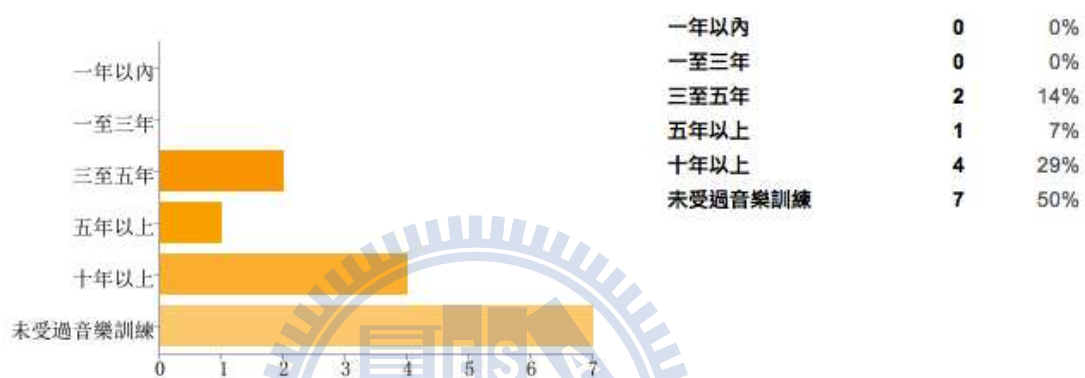


圖 5.1 受試者受過音樂訓練時間比例圖

接著我們調查接觸過非調性音樂者的比例，其中有八位受試者接觸過非調性音樂，比例為 57%，沒接觸過的受試者為六位，比例為 43%，未受過音樂訓練的受試者中 (7 位)，有一位接觸過非調性音樂，比例為 14%。圖 5.2 為受試者接觸過非調性音樂的比例圖。



圖 5.2 受試者接觸過非調性音樂比例圖

實驗中，我們提供八首由系統產生出來的音樂 MIDI 檔，其中有四首為非調性，四首為調性音樂，調性音樂中有三首為大調，一首為小調；非調性音樂中有兩首為 3/4 拍，兩首為 4/4 拍，調性音樂亦同。長度方面，十六小節的有三首，八小節為五首，由於超過十六小節以上，並不容易聽出動機與主題變化，因此我們最長只提供到十六小節。

首先是調性辨識率，我們採詢問是否為有調音樂的問法來調查，若受試者回答同意，則表示該編號 MIDI 檔為有調音樂，同意的選項中又分為大調及小調；若回答不同意，則表示為非調性音樂。測試檔編號 01、04、06 及 08 為有調音樂，編號 02、03、05 及 07 為非調音樂。表 5.1 為調性辨識率統計表，其中，有調的調性辨識率平均為 78.25%，非調性的調性辨識率平均為 52.75%。在有調的調性辨識率中，編號 01、04 及 08 為大調，其認為是大調的比例分別為 82%、82% 及 80%，編號 06 為小調，認為是小調的比例為 67%。

表 5.1 八首 MIDI 檔的調性辨識率統計表

| | 認為 編號 01 是有調 音樂 | 認為 編號 02 是有調 音樂 | 認為 編號 03 是有調 音樂 | 認為 編號 04 是有調 音樂 | 認為 編號 05 是有調 音樂 | 認為 編號 06 是有調 音樂 | 認為 編號 07 是有調 音樂 | 認為 編號 08 是有調 音樂 |
|---------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| 同意 (大調) | 65% (9 人) | 14% (2 人) | 0% (0 人) | 65% (9 人) | 7% (1 人) | 29% (4 人) | 7% (1 人) | 58% (8 人) |
| 同意 (小調) | 14% (2 人) | 43% (6 人) | 21% (3 人) | 14% (2 人) | 36% (5 人) | 57% (8 人) | 21% (3 人) | 14% (2 人) |
| 不同意 (非調) | 14% (2 人) | 29% (4 人) | 72% (10 人) | 7% (1 人) | 43% (6 人) | 7% (1 人) | 64% (9 人) | 21% (3 人) |
| 沒意見 (聽不出來) | 7% (1 人) | 14% (2 人) | 7% (1 人) | 14% (2 人) | 14% (2 人) | 7% (1 人) | 7% (1 人) | 7% (1 人) |

對於未受過音樂訓練的受試者，在實驗之前，我們提供非調性、大調及小調的範例音樂片段各三首，其中，非調性範例音樂為荀白克弦樂四重奏 No. 4 第一樂章，大調範例音樂為巴哈平均律 BWV 846 前奏曲，為 C 大調；小調範例音樂為巴哈平均律 BWV 847 前奏曲，為 c 小調。針對 14 位受試者中，未受過音樂訓練的七位受試者的結果，整理出表 5.2。

從表中可看出，非調性以編號 02 及 07 辨識效果最好，辨識率為 43%，調性音樂以編號 01 及 04 為最好，大調辨識率為 43%。調性整體平均辨識率為 61%，非調性的辨識率只有 28.5%，顯示未受過音樂訓練者對於非調性的辨識程度不高。

表 5.2 未受過音樂訓練受試者的八首 MIDI 檔的調性辨識率統計表

| | 認為 編號 01 是有調 音樂 | 認為 編號 02 是有調 音樂 | 認為 編號 03 是有調 音樂 | 認為 編號 04 是有調 音樂 | 認為 編號 05 是有調 音樂 | 認為 編號 06 是有調 音樂 | 認為 編號 07 是有調 音樂 | 認為 編號 08 是有調 音樂 |
|---------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| 同意 (大調) | 43% (3 人) | 14% (1 人) | 0% (0 人) | 43% (3 人) | 14% (1 人) | 43% (3 人) | 14% (1 人) | 29% (2 人) |
| 同意 (小調) | 14% (1 人) | 72% (5 人) | 29% (2 人) | 14% (1 人) | 43% (3 人) | 29% (2 人) | 29% (2 人) | 29% (2 人) |
| 不同意 (非調) | 29% (2 人) | 0% (0 人) | 57% (4 人) | 14% (1 人) | 14% (1 人) | 14% (1 人) | 43% (3 人) | 29% (2 人) |
| 沒意見 (聽不出來) | 14% (1 人) | 14% (1 人) | 14% (1 人) | 29% (2 人) | 29% (2 人) | 14% (1 人) | 14% (1 人) | 14% (1 人) |

在動機變化程度方面，我們以編號 04 及編號 07 的 MIDI 為例子，這兩首在實作結果時已有展示。我們使用 Likert 5-point Scale 的方法計算平均分數，將選項分為五個等級：非常明顯為五分，明顯為四分，普通為三分，不明顯為二分，非常不明顯為一分。圖 4.6 為編號 04 的 MIDI 檔譜例，紅色框框表示動機及動機變化的部分，我們讓受試者聽完這首 MIDI 檔並詢問動機變化程度，其結果可從表 5.3 中看出，動機辨識率的平均分數為 3.5。

圖 4.5 為編號 07 的 MIDI 檔譜例，紅色框框為動機變化部分，此首 MIDI 的動機變化皆為 Retrograde，由於有 A 及 B 兩個主題，受試者較不易聽出各自的動機變化，因此辨識率較編號 06 的 MIDI 檔低，為 2.79，可由表 5.2 中看出其平均。

表 5.3 八首 MIDI 檔的動機變化明顯程度統計表

| | 認為 編號 01 動機變 化明顯 | 認為 編號 02 動機變 化明顯 | 認為 編號 03 動機變 化明顯 | 認為 編號 04 動機變 化明顯 | 認為 編號 05 動機變 化明顯 | 認為 編號 06 動機變 化明顯 | 認為 編號 07 動機變 化明顯 | 認為 編號 08 動機變 化明顯 |
|--------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|
| 個數 有效的 | 14 | 14 | 14 | 14 | 14 | 14 | 14 | 14 |
| 遺漏值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 平均數 | 3.36 | 3.64 | 3.43 | 3.5 | 3.29 | 3.71 | 2.79 | 2.93 |
| 眾數 | 3 ^a | 4 | 4 | 3 ^a | 4 | 4 | 3 | 2 |
| 標準差 | 1.336 | .745 | 1.399 | .941 | .994 | 1.069 | .893 | .917 |

*a. 存在多個眾數，顯示的為最小值。

我們針對未受過音樂訓練的受試者做統計，結果如表 5.4，實驗提供貝多芬命運交響曲為動機範例，讓受試者先聽過後再做實驗，整體平均為 2.96。

表 5.4 未受過音樂訓練受試者的八首 MIDI 檔的動機變化明顯程度統計表

| | | 認為 編號 01 動機變 化明顯 | 認為 編號 02 動機變 化明顯 | 認為 編號 03 動機變 化明顯 | 認為 編號 04 動機變 化明顯 | 認為 編號 05 動機變 化明顯 | 認為 編號 06 動機變 化明顯 | 認為 編號 07 動機變 化明顯 | 認為 編號 08 動機變 化明顯 |
|----|-----|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|
| 個數 | 有效的 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| | 遺漏值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 平均數 | 2.43 | 3.71 | 2.71 | 3.14 | 3.29 | 3.14 | 2.43 | 2.86 |
| | 眾數 | 2 | 4 | 1 ^a | 3 | 4 | 3 ^a | 3 | 2 |
| | 標準差 | .976 | .488 | 1.604 | 1.069 | 1.253 | 1.069 | .787 | 1.069 |

*a. 存在多個眾數，顯示的為最小值。

最後是音樂的喜好程度，實驗將喜好程度分為五等分，五為喜好程度最高，一則表示最低，表 5.5 為八首 MIDI 檔的喜好程度統計表。我們用 Likert 5-point Scale 的方法計算，結果顯示，大部分受試者對於有調音樂的喜好程度大於非調性音樂，有調音樂編號為 01、04、06、08，喜好程度平均分數為 3.29，非調音樂編號為 02、03、05、07，喜好程度平均分數只有 2.66。

表 5.5 八首 MIDI 檔的喜好程度統計表

| | | 編號 01 的喜 好程度 | 編號 02 的喜 好程度 | 編號 03 的喜 好程度 | 編號 04 的喜 好程度 | 編號 05 的喜 好程度 | 編號 06 的喜 好程度 | 編號 07 的喜 好程度 | 編號 08 的喜 好程度 |
|----|-----|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| 個數 | 有效的 | 14 | 14 | 14 | 14 | 14 | 14 | 14 | 14 |
| | 遺漏值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 平均數 | 3.36 | 2.86 | 2.29 | 3.29 | 3.07 | 3.57 | 2.43 | 2.93 |
| | 眾數 | 4 | 3 | 2 | 3 | 3 | 3 | 3 | 3 |
| | 標準差 | 1.008 | .770 | .726 | .825 | .616 | .851 | .851 | .917 |

*a. 存在多個眾數，顯示的為最小值。

第六章、結論與未來研究

6.1 結論

本論文中，我們以音高類集為理論基礎來作自動作曲的研究。在結構上，系統達到動機變化及主題發展的功能，動機的長度為兩小節，由一組特定的節奏及音高組成，主題則包含動機及動機變化，長度為八小節，我們提供簡單的曲式以三段式為例 ABA'，亦即主題 A 為第一至第八小節，第九至十六小節為主題 B，接著是主題 A 發展；在有調音樂中，除了 24 個大小調外，還加入終止式以加強和聲結構；在節奏部分，我們嘗試加入複雜度的概念來增加各種節奏的組合；而最重要的音高部分，為了增加其變化，我們使用音高類集理論中的 Prime Form，其變化有 Transposition、Inversion 及 T_nI 共 24 種，在無調性音樂中，Prime Form 的 24 種變化皆可使用，但到了有調音樂中，為了符合調性音樂的特性，系統使用 filter 機制來篩選出可用的 Prime Form 組合，再搭配終止式而成。

而根據三個實驗的結果，有調的調性辨識率平均為 78.25%，非調性的調性辨識率平均為 52.75%，表示非調性音樂較不容易辨識；動機辨識率及喜好程度的平均分數以 Likert 5-point Scale 計算，動機辨識率平均分數為 3.5，結果在平均之上；喜好程度的整體平均分數為 2.975，其中，有調音喜好程度的平均分數為 3.29，非調音樂喜好程度的平均分數為 2.66，顯示大部分受試者對於系統產生的有調音樂是可接受的。

6.2 未來研究

在未來的研究中，我們考慮加入三個方向來改進本論文的系統：

一、風格分析：現在的系統只具備基礎的自動作曲功能，因此曲風聽起來都是差不多的，若能加入風格的選項，則能增加更多可能性，因此可以藉由分析大量不同風格的樂曲，歸納出不同種風格的音高組成、節奏變化、和聲搭配及結構規則等，如此便可產生變化更多的樂曲。

二、加入更多樂理分析：系統產生出的樂曲中，對於音與音之間的跳動程度並無加以研究，例如：Leap、Stepwise、Arpeggio 等，若能分析以上三者樂曲中出現的比例，並加入系統中加以運用，必能在旋律的流暢度上有更多幫助。

三、使用馬可夫鏈：由於本系統在和聲方面只使用終止式，對於和聲的變化不大，因此在未來考慮加入馬可夫鏈，來增加和聲變化的可能性，也可產生出更多樣的樂曲。

參考文獻

- [1] Ames, C., “The Markov Process as a Compositional Model: A Survey and Tutorial”, Leonardo, vol. 22, no. 2, pp. 175—187, 1989.
- [2] Cambouropoulos, E., “Markov Chains as an Aid to Computer Assisted Composition”, Musical Praxis, vol. 1, no. 1, pp. 41—52, 1994.
- [3] Farbood, M., Schoner, B., “Analysis and Synthesis of Palestrina-Style Counterpoint Using Markov Chains”, the International Computer Music Conference, September in Havana, Cuba, pp. 18—22, 2001.
- [4] Wiggins, G., Papadopoulos, G., Phon-Amnuaisuk S., Tuson, A., “Evolutionary Methods for Musical Composition”, Partial Proceedings of the 2nd International Conference CASYS’98 on Computing Anticipatory Systems, Liège, Belgium, pp. 10—14, 1998.
- [5] Moroni, A., Manzolli, J., Zuben, F. V., Gudwin, R., “Evolutionary Computation applied to Algorithmic Composition”, Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on, 1999.
- [6] Povel, D.J., “Melody Generator: A Device for Algorithmic Music Construction”, Journal of Software Engineering & Applications, 3, 683-695, 2010.
- [7] Temperley, D., “A Probabilistic Model of Melody Perception”, Cognitive Science, vol.32, no.2, pp418—444, 2008.
- [8] 張錦鴻，基礎樂理，大陸書店，民國八十五年。
- [9] Straus, J.N., Introduction to Post-Tonal Theory, Prentice Hall, Englewood Cliffs, New Jersey, 1990.
- [10] Schuijjer, M., Analyzing Atonal Music: Pitch-Class Set Theory and Its Contexts, Eastman Studies in Music 60. Rochester, NY: University of Rochester Press, 2008.
- [11] Nelson, P., Pitch Class Sets, Revised on January 20 2007, <http://composertools.com/Theory/PCSets/>, accessed April 08, 2011.
- [12] Schoenberg, A., Stein, L., Fundamentals of Music Composition, faber and faber, LONDON · BOSTON, 1967.
- [13] Stein, L.: Structure & Style: The Study and Analysis of Musical Forms, Summy-Birchard Music, 1979.
- [14] 黎翁斯坦 (Leon Stein) 著，音樂的結構與風格，康謳主譯，全音樂譜出版社，1974。
- [15] Gehrkens, K.W., The Project Gutenberg eBook, Music Notation and Terminology, 2006.
- [16] Scruton, R., The Aesthetics of Music, Oxford: Clarendon Press, 1997.
- [17] Thul, E., Toussaint, G. T., “Rhythm complexity measures: A comparison of mathematical models of human perception and performance”, In Proc. 9th International Conference on Music Information Retrieval, Philadelphia, USA, September, pp. 14—18, 2008.
- [18] Shmulevich, I., Povel, D.-J., Complexity measures of musical rhythms', in Rhythm Perception and Production, Eds. Desain, P. and Windsor, W.L. (Lisse: Swets & Zeitlinger), pp. 239--244, 2000.
- [19] Longuet-Higgins, H., Lee, C., “The rhythmic interpretation of monophonic music”, Music Perception, Vol. 1, No. 4, pp. 424—441, 1984.
- [20] Fitch, W. T., Rosenfeld, A. J.: Perception and production of syncopated rhythms, Music Perception, Vol. 25, No. 1, pp. 43—58, 2007.

- [21] Smith, L., Honing, H.: Evaluating and extending computational models of rhythmic syncopation in music, in Proceedings of the International Computer Music Conference, pp. 688—691, 2006.
- [22] Thul, E., Toussaint, G.T.: “Analysis of musical rhythm complexity measures in a cultural context”. In Proceedings of the 2008 C3S2E conference (C3S2E '08). ACM, New York, NY, USA, pp. 1--9, 2008.
- [23] Thul, E., Measuring the Complexity of Musical Rhythm, McGill University Montreal, Quebec, 2008.
- [24] Cope, D., The Algorithmic Composer, Madison, Wisconsin ,2000.
- [25] Morris, R. Composition with Pitch-class : A Theory of Compositional Design, Vali-Ballou Press, USA, 1987.



附錄：The Prim Forms Table (資料來源：<http://composertools.com/Theory/>)

| Interval Vector | count | Forte code | Prime form | Inverted form |
|---|-------|------------|---------------------|---------------|
| Sets of 0 pitch classes, 0 intervals (1 vector, 1 quality, 1 total) | | | | |
| <000000> | (1) | | () {silence} | |
| Sets of 1 pitch classes, 0 intervals (1 vector, 1 quality, 12 total) | | | | |
| <000000> | (12) | | (0) {single-note} | |
| Sets of 2 pitch classes, 1 intervals (6 vectors, 6 qualities, 66 total) | | | | |
| <100000> | (12) | | (0,1) {half-step} | |
| <010000> | (12) | | (0,2) {whole-step} | |
| <001000> | (12) | | (0,3) {minor-third} | |
| <000100> | (12) | | (0,4) {major-third} | |
| <000010> | (12) | | (0,5) {perfect} | |
| <000001> | (6) | | (0,6) {tritone} | |
| Sets of 3 pitch classes, 3 intervals (12 vectors, 19 qualities, 220 total) | | | | |
| <210000> | (12) | 3-1: | (0,1,2) | |
| <111000> | (24) | 3-2: | (0,1,3) | [0,2,3] |
| <101100> | (24) | 3-3: | (0,1,4) | [0,3,4] |
| <100110> | (24) | 3-4: | (0,1,5) | [0,4,5] |
| <100011> | (24) | 3-5: | (0,1,6) | [0,5,6] |
| <020100> | (12) | 3-6: | (0,2,4) | |
| <011010> | (24) | 3-7: | (0,2,5) | [0,3,5] |
| <010101> | (24) | 3-8: | (0,2,6) {It.} | [0,4,6] |
| <010020> | (12) | 3-9: | (0,2,7) {quar-3} | |
| <002001> | (12) | 3-10: | (0,3,6) {dim} | |
| <001110> | (24) | 3-11: | (0,3,7) {min} | [0,4,7] {maj} |
| <000300> | (4) | 3-12: | (0,4,8) {aug} | |
| Sets of 4 pitch classes, 6 intervals (28 vectors, 43 qualities, 495 total) | | | | |
| <321000> | (12) | 4-1: | (0,1,2,3) | |
| <221100> | (24) | 4-2: | (0,1,2,4) | [0,2,3,4] |
| <212100> | (12) | 4-3: | (0,1,3,4) | |
| <211110> | (24) | 4-4: | (0,1,2,5) | [0,3,4,5] |
| <210111> | (24) | 4-5: | (0,1,2,6) | [0,4,5,6] |
| <210021> | (12) | 4-6: | (0,1,2,7) | |
| <201210> | (12) | 4-7: | (0,1,4,5) | |
| <200121> | (12) | 4-8: | (0,1,5,6) | |
| <200022> | (6) | 4-9: | (0,1,6,7) | |
| <122010> | (12) | 4-10: | (0,2,3,5) | |
| <121110> | (24) | 4-11: | (0,1,3,5) | [0,2,4,5] |
| <112101> | (24) | 4-12: | (0,2,3,6) | [0,3,4,6] |
| <112011> | (24) | 4-13: | (0,1,3,6) | [0,3,5,6] |
| <111120> | (24) | 4-14: | (0,2,3,7) | [0,4,5,7] |
| <111111> | (48) | 4-Z15: | (0,1,4,6) | [0,2,5,6] |
| | | 4-Z29: | (0,1,3,7) | [0,4,6,7] |

| Interval Vector | count | Forte code | Prime form | Inverted form |
|--|-------|------------|----------------------|-----------------|
| <110121> | (24) | 4-16: | (0,1,5,7) | [0,2,6,7] |
| <102210> | (12) | 4-17: | (0,3,4,7) | |
| <102111> | (24) | 4-18: | (0,1,4,7) | [0,3,6,7] |
| <101310> | (24) | 4-19: | (0,1,4,8){mM7} | [0,3,4,8] |
| <101220> | (12) | 4-20: | (0,1,5,8){maj7} | |
| <030201> | (12) | 4-21: | (0,2,4,6) | |
| <021120> | (24) | 4-22: | (0,2,4,7) | [0,3,5,7] |
| <021030> | (12) | 4-23: | (0,2,5,7){quar-4} | |
| <020301> | (12) | 4-24: | (0,2,4,8){7+5} | |
| <020202> | (6) | 4-25: | (0,2,6,8){fr.,7-5} | |
| <012120> | (12) | 4-26: | (0,3,5,8){min7,maj6} | |
| <012111> | (24) | 4-27: | (0,2,5,8){hd7} | [0,3,6,8]{dom7} |
| <004002> | (3) | 4-28: | (0,3,6,9){dd7} | |
| Sets of 5 pitch classes, 10 intervals (35 vectors, 66 qualities, 792 total) | | | | |
| <432100> | (12) | 5-1: | (0,1,2,3,4) | |
| <332110> | (24) | 5-2: | (0,1,2,3,5) | [0,2,3,4,5] |
| <322210> | (24) | 5-3: | (0,1,2,4,5) | [0,1,3,4,5] |
| <322111> | (24) | 5-4: | (0,1,2,3,6) | [0,3,4,5,6] |
| <321121> | (24) | 5-5: | (0,1,2,3,7) | [0,4,5,6,7] |
| <311221> | (24) | 5-6: | (0,1,2,5,6) | [0,1,4,5,6] |
| <310132> | (24) | 5-7: | (0,1,2,6,7) | [0,1,5,6,7] |
| <232201> | (12) | 5-8: | (0,2,3,4,6) | |
| <231211> | (24) | 5-9: | (0,1,2,4,6) | [0,2,4,5,6] |
| <223111> | (24) | 5-10: | (0,1,3,4,6) | [0,2,3,5,6] |
| <222220> | (24) | 5-11: | (0,2,3,4,7) | [0,3,4,5,7] |
| <222121> | (36) | 5-Z12: | (0,1,3,5,6) | |
| | | 5-Z36: | (0,1,2,4,7) | [0,3,5,6,7] |
| <221311> | (24) | 5-13: | (0,1,2,4,8) | [0,2,3,4,8] |
| <221131> | (24) | 5-14: | (0,1,2,5,7) | [0,2,5,6,7] |
| <220222> | (12) | 5-15: | (0,1,2,6,8) | |
| <213211> | (24) | 5-16: | (0,1,3,4,7) | [0,3,4,6,7] |
| <212320> | (24) | 5-Z17: | (0,1,3,4,8) | |
| | | 5-Z37: | (0,3,4,5,8) | |
| <212221> | (48) | 5-Z18: | (0,1,4,5,7) | [0,2,3,6,7] |
| | | 5-Z38: | (0,1,2,5,8) | [0,3,6,7,8] |
| <212122> | (24) | 5-19: | (0,1,3,6,7) | [0,1,4,6,7] |
| <211231> | (24) | 5-20: | (0,1,3,7,8) | [0,1,5,7,8] |
| <202420> | (24) | 5-21: | (0,1,4,5,8) | [0,3,4,7,8] |
| <202321> | (12) | 5-22: | (0,1,4,7,8) | |
| <132130> | (24) | 5-23: | (0,2,3,5,7) | [0,2,4,5,7] |
| <131221> | (24) | 5-24: | (0,1,3,5,7) | [0,2,4,6,7] |
| <123121> | (24) | 5-25: | (0,2,3,5,8) | [0,3,5,6,8] |
| <122311> | (24) | 5-26: | (0,2,4,5,8) | [0,3,4,6,8] |

| Interval Vector | count | Forte code | Prime form | Inverted form |
|--|-------|------------|--------------------------------|-------------------|
| <122230> | (24) | 5-27: | (0,1,3,5,8) | [0,3,5,7,8]{min9} |
| <122212> | (24) | 5-28: | (0,2,3,6,8) | [0,2,5,6,8] |
| <122131> | (24) | 5-29: | (0,1,3,6,8) | [0,2,5,7,8] |
| <121321> | (24) | 5-30: | (0,1,4,6,8) | [0,2,4,7,8] |
| <114112> | (24) | 5-31: | (0,1,3,6,9) | [0,2,3,6,9]{7-9} |
| <113221> | (24) | 5-32: | (0,1,4,6,9) | [0,2,5,6,9]{7+9} |
| <040402> | (12) | 5-33: | (0,2,4,6,8){9+5,9-5} | |
| <032221> | (12) | 5-34: | (0,2,4,6,9){dom9} | |
| <032140> | (12) | 5-35: | (0,2,4,7,9){pentatonic,Quar-5} | |
| Sets of 6 pitch classes, 15 intervals (35 vectors, 80 qualities, 924 total) | | | | |
| <543210> | (12) | 6-1: | (0,1,2,3,4,5) | |
| <443211> | (24) | 6-2: | (0,1,2,3,4,6) | [0,2,3,4,5,6] |
| <433221> | (48) | 6-Z3: | (0,1,2,3,5,6) | [0,1,3,4,5,6] |
| | | 6-Z36: | (0,1,2,3,4,7) | [0,3,4,5,6,7] |
| <432321> | (24) | 6-Z4: | (0,1,2,4,5,6) | |
| | | 6-Z37: | (0,1,2,3,4,8) | |
| <422232> | (24) | 6-5: | (0,1,2,3,6,7) | [0,1,4,5,6,7] |
| <421242> | (24) | 6-Z6: | (0,1,2,5,6,7) | |
| | | 6-Z38: | (0,1,2,3,7,8) | |
| <420243> | (6) | 6-7: | (0,1,2,6,7,8) | |
| <343230> | (12) | 6-8: | (0,2,3,4,5,7) | |
| <342231> | (24) | 6-9: | (0,1,2,3,5,7) | [0,2,4,5,6,7] |
| <333321> | (48) | 6-Z10: | (0,1,3,4,5,7) | [0,2,3,4,6,7] |
| | | 6-Z39: | (0,2,3,4,5,8) | [0,3,4,5,6,8] |
| <333231> | (48) | 6-Z11: | (0,1,2,4,5,7) | [0,2,3,5,6,7] |
| | | 6-Z40: | (0,1,2,3,5,8) | [0,3,5,6,7,8] |
| <332232> | (48) | 6-Z12: | (0,1,2,4,6,7) | [0,1,3,5,6,7] |
| | | 6-Z41: | (0,1,2,3,6,8) | [0,2,5,6,7,8] |
| <324222> | (24) | 6-Z13: | (0,1,3,4,6,7) | |
| | | 6-Z42: | (0,1,2,3,6,9) | |
| <323430> | (24) | 6-14: | (0,1,3,4,5,8) | [0,3,4,5,7,8] |
| <323421> | (24) | 6-15: | (0,1,2,4,5,8) | [0,3,4,6,7,8] |
| <322431> | (24) | 6-16: | (0,1,4,5,6,8) | [0,2,3,4,7,8] |
| <322332> | (48) | 6-Z17: | (0,1,2,4,7,8) | [0,1,4,6,7,8] |
| | | 6-Z43: | (0,1,2,5,6,8) | [0,2,3,6,7,8] |
| <322242> | (24) | 6-18: | (0,1,2,5,7,8) | [0,1,3,6,7,8] |
| <313431> | (48) | 6-Z19: | (0,1,3,4,7,8) | [0,1,4,5,7,8] |
| | | 6-Z44: | (0,1,2,5,6,9) | [0,1,4,5,6,9] |
| <303630> | (4) | 6-20: | (0,1,4,5,8,9) | |
| <242412> | (24) | 6-21: | (0,2,3,4,6,8) | [0,2,4,5,6,8] |
| <241422> | (24) | 6-22: | (0,1,2,4,6,8) | [0,2,4,6,7,8] |
| <234222> | (24) | 6-Z23: | (0,2,3,5,6,8) | |
| | | 6-Z45: | (0,2,3,4,6,9) | |
| <233331> | (48) | 6-Z24: | (0,1,3,4,6,8) | [0,2,4,5,7,8] |
| | | 6-Z46: | (0,1,2,4,6,9) | [0,2,4,5,6,9] |

| Interval Vector | count | Forte code | Prime form | Inverted form |
|--|-------|------------|-----------------------------|----------------------|
| <233241> | (48) | 6-Z25: | (0,1,3,5,6,8) | [0,2,3,5,7,8] |
| | | 6-Z47: | (0,1,2,4,7,9) | [0,2,3,4,7,9] |
| <232341> | (24) | 6-Z26: | (0,1,3,5,7,8) | |
| | | 6-Z48: | (0,1,2,5,7,9) | |
| <225222> | (24) | 6-27: | (0,1,3,4,6,9) | [0,2,3,5,6,9] |
| <224322> | (24) | 6-Z28: | (0,1,3,5,6,9) | |
| | | 6-Z49: | (0,1,3,4,7,9) | |
| <224232> | (24) | 6-Z29: | (0,1,3,6,8,9) | |
| | | 6-Z50: | (0,1,4,6,7,9) | |
| <224223> | (12) | 6-30: | (0,1,3,6,7,9) | [0,2,3,6,8,9] |
| <223431> | (24) | 6-31: | (0,1,3,5,8,9) | [0,1,4,6,8,9] |
| <143250> | (12) | 6-32: | (0,2,4,5,7,9){min11} | |
| <143241> | (24) | 6-33: | (0,2,3,5,7,9) | [0,2,4,6,7,9]{dom11} |
| <142422> | (24) | 6-34: | (0,1,3,5,7,9) | [0,2,4,6,8,9] |
| <060603> | (2) | 6-35: | (0,2,4,6,8,A){wholetone} | |
| Sets of 7 pitch classes, 21 intervals (35 vectors, 66 qualities, 792 total) | | | | |
| <654321> | (12) | 7-1: | (0,1,2,3,4,5,6) | |
| <554331> | (24) | 7-2: | (0,1,2,3,4,5,7) | [0,2,3,4,5,6,7] |
| <544431> | (24) | 7-3: | (0,1,2,3,4,5,8) | [0,3,4,5,6,7,8] |
| <544332> | (24) | 7-4: | (0,1,2,3,4,6,7) | [0,1,3,4,5,6,7] |
| <543342> | (24) | 7-5: | (0,1,2,3,5,6,7) | [0,1,2,4,5,6,7] |
| <533442> | (24) | 7-6: | (0,1,2,3,4,7,8) | [0,1,4,5,6,7,8] |
| <532353> | (24) | 7-7: | (0,1,2,3,6,7,8) | [0,1,2,5,6,7,8] |
| <454422> | (12) | 7-8: | (0,2,3,4,5,6,8) | |
| <453432> | (24) | 7-9: | (0,1,2,3,4,6,8) | [0,2,4,5,6,7,8] |
| <445332> | (24) | 7-10: | (0,1,2,3,4,6,9) | [0,2,3,4,5,6,9] |
| <444441> | (24) | 7-11: | (0,1,3,4,5,6,8) | [0,2,3,4,5,7,8] |
| <444342> | (36) | 7-Z12: | (0,1,2,3,4,7,9) | |
| | | 7-Z36: | (0,1,2,3,5,6,8) | [0,2,3,5,6,7,8] |
| <443532> | (24) | 7-13: | (0,1,2,4,5,6,8) | [0,2,3,4,6,7,8] |
| <443352> | (24) | 7-14: | (0,1,2,3,5,7,8) | [0,1,3,5,6,7,8] |
| <442443> | (12) | 7-15: | (0,1,2,4,6,7,8) | |
| <435432> | (24) | 7-16: | (0,1,2,3,5,6,9) | [0,1,3,4,5,6,9] |
| <442443> | (12) | 7-15: | (0,1,2,4,6,7,8) | |
| <435432> | (24) | 7-16: | (0,1,2,3,5,6,9) | [0,1,3,4,5,6,9] |
| <434541> | (24) | 7-Z17: | (0,1,2,4,5,6,9) | |
| | | 7-Z37: | (0,1,3,4,5,7,8) | |
| <434442> | (48) | 7-Z18: | (0,1,4,5,6,7,9) | [0,2,3,4,5,8,9] |
| | | 7-Z38: | (0,1,2,4,5,7,8) | [0,1,3,4,6,7,8] |
| <434343> | (24) | 7-19: | (0,1,2,3,6,7,9) | [0,1,2,3,6,8,9] |
| <433452> | (24) | 7-20: | (0,1,2,4,7,8,9) | [0,1,2,5,7,8,9] |
| <424641> | (24) | 7-21: | (0,1,2,4,5,8,9) | [0,1,3,4,5,8,9] |
| <424542> | (12) | 7-22: | (0,1,2,5,6,8,9){hungar-min} | |
| <354351> | (24) | 7-23: | (0,2,3,4,5,7,9) | [0,2,4,5,6,7,9] |
| <353442> | (24) | 7-24: | (0,1,2,3,5,7,9) | [0,2,4,6,7,8,9] |

| Interval Vector | count | Forte code | Prime form | Inverted form |
|--|--------------|-------------------|------------------------------|----------------------|
| <345342> | (24) | 7-25: | (0,2,3,4,6,7,9) | [0,2,3,5,6,7,9] |
| <344532> | (24) | 7-26: | (0,1,3,4,5,7,9) | [0,2,4,5,6,8,9] |
| <344451> | (24) | 7-27: | (0,1,2,4,5,7,9) | [0,2,4,5,7,8,9] |
| <344433> | (24) | 7-28: | (0,1,3,5,6,7,9) | [0,2,3,4,6,8,9] |
| <344352> | (24) | 7-29: | (0,1,2,4,6,7,9) | [0,2,3,5,7,8,9] |
| <343542> | (24) | 7-30: | (0,1,2,4,6,8,9) | [0,1,3,5,7,8,9] |
| <336333> | (24) | 7-31: | (0,1,3,4,6,7,9) | [0,2,3,5,6,8,9] |
| <335442> | (24) | 7-32: | (0,1,3,4,6,8,9){harm-min} | [0,1,3,5,6,8,9] |
| <262623> | (12) | 7-33: | (0,1,2,4,6,8,A) | |
| <254442> | (12) | 7-34: | (0,1,3,4,6,8,A) | |
| <254361> | (12) | 7-35: | (0,1,3,5,6,8,A){diatonic} | |
| Sets of 8 pitch classes, 28 intvls (28 vectors, 43 qualities, 495 total) | | | | |
| <765442> | (12) | 8-1: | (0,1,2,3,4,5,6,7) | |
| <665542> | (24) | 8-2: | (0,1,2,3,4,5,6,8) | [0,2,3,4,5,6,7,8] |
| <656542> | (12) | 8-3: | (0,1,2,3,4,5,6,9) | |
| <655552> | (24) | 8-4: | (0,1,2,3,4,5,7,8) | [0,1,3,4,5,6,7,8] |
| <654553> | (24) | 8-5: | (0,1,2,3,4,6,7,8) | [0,1,2,4,5,6,7,8] |
| <654463> | (12) | 8-6: | (0,1,2,3,5,6,7,8) | |
| <645652> | (12) | 8-7: | (0,1,2,3,4,5,8,9) | |
| <644563> | (12) | 8-8: | (0,1,2,3,4,7,8,9) | |
| <644464> | (6) | 8-9: | (0,1,2,3,6,7,8,9) | |
| <566452> | (12) | 8-10: | (0,2,3,4,5,6,7,9) | |
| <565552> | (24) | 8-11: | (0,1,2,3,4,5,7,9) | [0,2,4,5,6,7,8,9] |
| <556543> | (24) | 8-12: | (0,1,3,4,5,6,7,9) | [0,2,3,4,5,6,8,9] |
| <556453> | (24) | 8-13: | (0,1,2,3,4,6,7,9) | [0,2,3,5,6,7,8,9] |
| <555562> | (24) | 8-14: | (0,1,2,4,5,6,7,9) | [0,2,3,4,5,7,8,9] |
| <555553> | (48) | 8-Z15: | (0,1,2,3,4,6,8,9) | [0,1,3,5,6,7,8,9] |
| | | 8-Z29: | (0,1,2,3,5,6,7,9) | [0,2,3,4,6,7,8,9] |
| <554563> | (24) | 8-16: | (0,1,2,3,5,7,8,9) | [0,1,2,4,6,7,8,9] |
| <546652> | (12) | 8-17: | (0,1,3,4,5,6,8,9) | |
| <546553> | (24) | 8-18: | (0,1,2,3,5,6,8,9) | [0,1,3,4,6,7,8,9] |
| <545752> | (24) | 8-19: | (0,1,2,4,5,6,8,9) | [0,1,3,4,5,7,8,9] |
| <545662> | (12) | 8-20: | (0,1,2,4,5,7,8,9) | |
| <474643> | (12) | 8-21: | (0,1,2,3,4,6,8,A) | |
| <465562> | (24) | 8-22: | (0,1,2,3,5,6,8,A) | [0,1,3,4,5,6,8,A] |
| <465472> | (12) | 8-23: | (0,1,2,3,5,7,8,A) | |
| <464743> | (12) | 8-24: | (0,1,2,4,5,6,8,A) | |
| <464644> | (6) | 8-25: | (0,1,2,4,6,7,8,A) | |
| <456562> | (12) | 8-26: | (0,1,2,4,5,7,9,A) | |
| <456553> | (24) | 8-27: | (0,1,2,4,5,7,8,A) | [0,1,3,4,6,7,8,A] |
| <448444> | (3) | 8-28: | (0,1,3,4,6,7,9,A){octatonic} | |
| Sets of 9 pitch classes, 36 intervals (12 vectors, 19 qualities, 220 total) | | | | |
| <876663> | (12) | 9-1: | (0,1,2,3,4,5,6,7,8) | |
| <777663> | (24) | 9-2: | (0,1,2,3,4,5,6,7,9) | [0,2,3,4,5,6,7,8,9] |
| <767763> | (24) | 9-3: | (0,1,2,3,4,5,6,8,9) | [0,1,3,4,5,6,7,8,9] |

| Interval Vector | count | Forte code | Prime form | Inverted form |
|--|--------------|-------------------|--|----------------------|
| <766773> | (24) | 9-4: | (0,1,2,3,4,5,7,8,9) | [0,1,2,4,5,6,7,8,9] |
| <766674> | (24) | 9-5: | (0,1,2,3,4,6,7,8,9) | [0,1,2,3,5,6,7,8,9] |
| <686763> | (12) | 9-6: | (0,1,2,3,4,5,6,8,A) | |
| <677673> | (24) | 9-7: | (0,1,2,3,4,5,7,8,A) | [0,1,3,4,5,6,7,8,A] |
| <676764> | (24) | 9-8: | (0,1,2,3,4,6,7,8,A) | [0,1,2,4,5,6,7,8,A] |
| <676683> | (12) | 9-9: | (0,1,2,3,5,6,7,8,A) | |
| <668664> | (12) | 9-10: | (0,1,2,3,4,6,7,9,A) | |
| <667773> | (24) | 9-11: | (0,1,2,3,5,6,7,9,A) | [0,1,2,4,5,6,7,9,A] |
| <666963> | (4) | 9-12: | (0,1,2,4,5,6,8,9,A) | |
| Sets of 10 pitch classes, 45 intervals (6 vectors, 6 qualities, 66 total) | | | | |
| <988884> | (12) | | (0,1,2,3,4,5,6,7,8,9) | |
| <898884> | (12) | | (0,1,2,3,4,5,6,7,8,A) | |
| <889884> | (12) | | (0,1,2,3,4,5,6,7,9,A) | |
| <888984> | (12) | | (0,1,2,3,4,5,6,8,9,A) | |
| <888894> | (12) | | (0,1,2,3,4,5,7,8,9,A) | |
| <888885> | (6) | | (0,1,2,3,4,6,7,8,9,A) | |
| Sets of 11 pitch classes, 55 intervals (1 vector, 1 quality, 12 total) | | | | |
| <AAAAA5> | (12) | | (0,1,2,3,4,5,6,7,8,9,A) | |
| Sets of 12 pitch classes, 66 intervals (1 vector, 1 quality, 1 total) | | | | |
| <CCCCC6> | (1) | | (0,1,2,3,4,5,6,7,8,9,A,B) {chromatic} | |