國立交通大學

音樂研究所　音樂科技組

碩士論文

基於曲調輪廓能量守恆的
單聲部音高之演算法作曲

Algorithmic Composition of Monophonic Pitches Based on Conservation of Energy in Melodic Contour

研究生： 林欣名

指導教授： 黃志方、曾毓忠

中華民國一百年七月

基於曲調輪廓能量守恆的單聲部音高之演算法作曲

Algorithmic Composition of Monophonic Pitches Based on Conservation of Energy in Melodic Contour

研究生：林欣名　　Student: Hsin-Ming Lin

指導教授：黃志方、曾毓忠　　Advisors: Chih-Fang Huang and Yu-Chung Tseng

國立交通大學

音樂研究所　音樂科技組

碩士論文

A Thesis Submitted to
the Institute of Music
College of Humanities and Social Sciences
National Chiao Tung University
in Partial Fulfillment of
the Requirements for the Degree of
Master of Art
(Music Technology)

Hsinchu, Taiwan

July 2011

中華民國一百年七月

摘要

在眾多已提出的演算法作曲手段中，主音式的創作技巧與符號式的音級理論主宰了大部分的研究。然而，和聲乃由共生的曲調所構築；曲調線條才是西洋藝術音樂的根本法則。此外，當今的音樂科技，以低層內容為中心之方法與以高層後設資料為基礎之途徑間，兩者的橋樑正在萌芽。本論文從聲響頻率上的簡單比例出發，結合古典力學裡能量守恆的定律，再據以設計出相應的模型，來操縱曲調輪廓以及單旋律音高的運動。其描繪了音高間具體且可辨的能量比例，並反映了能量消耗與生成的物理現象。該創新模型能夠利用至符號式的記譜系統之外，以聲響頻率的原始資料作為輸出輸入。本研究實作出小型的視窗程式，能夠根據使用者參數自動產生出大致符合旋律創作之普遍性法則的單聲部連續音高。實驗結果展示出合理的曲調輪廓，亦即高度可唱的性質。未來期望能透過類似的能量概念，推導出適用於節奏與聲部間音程關係的演算法。

關鍵字：
演算法作曲、音高頻率、能量、曲調、輪廓

Abstract

In numerous methods of algorithmic composition of melody which have been proposed, techniques of homophonic composing and theories of symbolic pitch-class dominate most research. Nevertheless, harmony is constituted by symbiosis melodies; melodic lines are the fundamentals of western art music. Moreover, bridges between low-level content-centric methods and high-level metadata-based ones are just germinating in music technology nowadays. This thesis starts by the simple ratio of audio frequencies combining with the law of conservation of energy in classical dynamics and then devises a corresponding model in order to manipulate the melodic contour and motions of monophonic melody pitches. It represents the concrete and sensible energy ratio between each pitch as well as reflects the physical phenomenon of energy consumption and production. With the raw data of audio frequencies as input and output, that innovative model is able to be utilized beyond the system of symbolic notation. A small windows program is implemented in this research. It can automatically generate a series of monophonic pitches which roughly obeys the general rules of melody composing. Experimental results demonstrate reasonable melodic contour, i.e. a highly singable character. Through similar concepts of energy, an inference about the algorithm which is applicable to rhythms and relationship of intervals between voices could be expected in the future.

Keywords:
algorithmic composition, pitch frequency, energy, melody, contour

Table of Contents

List of Tables

List of Figures

1. Introduction

1.1 Rationale

In the initial sentence of *Romance of the Three Kingdoms*, a classic Chinese historical novel in the fourteenth century, its author pointed that this world must unite after lengthy separation and separate after lengthy union. As we can see, interdisciplinary research blooms in recent decades. “Music until the seventeenth century was one of the four mathematical disciplines of the quadrivium beside arithmetic, geometry, and astronomy” (Hsu, 1990, 938). Today, researchers weave the once divergent subjects together again. One of them is the algorithmic composition in music technology domain.

Since Lejaren A. Hiller and Leonard M. Isaacson’s “experimental music” (Hiller, 1959) and Iannis Xenakis’ “formalized music” (Xenakis, 1992), people invented and exploited numerous algorithmic composition methods. We could have a glimpse through many books. They convey valuable aspects of algorithmic composition: Robert Rowe distinguished between “symbolic processes” and “sub-symbolic processes” (Rowe, 2001, 1) in interactive music; Eduardo Reck Miranda divided algorithmic compositions into “microscopic level, note level, and building-block level” (Miranda, 2001, 2); Heinrich K. Taube exemplifies notes from “metalevel” (Taube, 2004); Gareth Loy provided comprehensive “mathematical foundations of music” (Loy, 2006) with applications to composition; Gerhard Nierhaus differentiated between “style imitations” and “genuine composition” (Nierhaus, 2009, 3). Those boundaries, however, are becoming more and more ambiguous in modern music technology. In fact, Rowe has demonstrated how to “bridge the levels between sub-symbolic and symbolic systems” (Rowe, 2009). In my opinion, to reconsider the phenomena of higher levels from the point of view which we have in lower levels can assist us in composing genuinely but with imitations by adopting the principles in old styles.

In the evolvement of western art music, the principle of the interaction between voices originates from the perspective of contrapuntal polyphonic music; the degree of consonance is measured by the vibration through physical presentation and auditory perception. Those two fundamentals form diverse music vocabularies and styles. They further develop into various complex or simplified theories. Even in the latter Equal Temperament instrument system and homophonic music style, the core thought of classical compositional methods is still unvaried. Nevertheless, since people invent and exploit music technology to engage in computer-aided composition, automated arrangement, and digital music information retrieval, symbolic pitch-class theories and homophonic composing techniques dominate most research. Although they have the convenience and bring respectable accomplishment, their spirit is essentially different from it in the art content which they imitate.

I have hungered for the underlying rules in melody and in the relations between simultaneous melodies. When I majored in theory and composition in the music department at university, my advisor spent months to teach me how to compose a melody. One of the most significant principle that he told me, is to manipulate ascending and descending movements of pitches like a pilot who keep the balance between speed and altitude of a very light aircraft. If he loses the speed, he will nosedive to regain it, and vice versa. That specific metaphor was apparent for me because I had taken honors physics during senior high school. Since then, I realized that some physical laws exist in arts, too. After I went to the graduate school, I saw more and more scientific ways to analyze, reorganize, devise, and actualize new sound and music. On account of my academic training, I intuitively associated the skill of melody pitches control with “kinetic energy” and “potential energy” (Thornton, 2004, 78) in classical dynamics.

My primary idea was to mathematically mimic the energy of horizontal melodic motion and vertical interval tension, from classical music composers’ perspectives originated in polyphonic vocal style and measured by the vibration through physical presentation and the auditory perception. Most prevalent methods of automated music generation treat notes on staff as “neutral pitch-classes” or “tonal pitch-classes” (Temperley, 2001, 118). In those kinds of system, all pitch interval degrees are generalized (e.g. Lewin, 1987; Morris, 1987; Krumhansl, 1990; Straus, 1990). Nevertheless, the energy consumption to make every diatonic pitch with the identical amplitude is an uneven distribution. It can be estimated by the ratio of audio frequencies, which is easy to obtain especially in terms of “Just Diatonic Scale” (Campbell, 1987, 172). With the values, the natural law of kinetic energy and potential energy may be applied to the motion of melody pitches.

1.2 Review

As far as musical pitches are concerned, people (e.g. Morris, 1998; Schell, 2002;

Tymoczko, 2008; Callender, 2008; Toussaint, 2010) usually refer to rules of harmony and voice leading (e.g. Kostka, 2000 and Aldwell, 2003). Moreover, Fred Lerdahl and Ray Jackendoff's famous "Generative Theory" (Lerdahl, 1983 and 2005) creates a hierarchical tree structure to vertically divide the melody and rank the pitches in each voice part. It is very effective in homophonic music. Unfortunately, it sacrificed a true melody's independence from chord construction and chord succession. They admit that "in truly contrapuntal music there is an important sense in which each line should receive its own separate structural description" (Lerdahl, 1983, 116). Actually, Llewelyn Southworth Lloyd and Hugh Boyle tell us about the connection between melody and harmony. "The melodic line was everything: harmony was in the making, it was being formed by writing concurrent melody" (Lloyd, 1979, 71). Hence, notwithstanding the fact that the melody composing is the most difficult genius to acquire and to instruct, it is indispensable for composers in order to accomplish elegant pieces no matter whether they intend to compose polyphonic music or not. Melody is the horizontal connection of pitches and rhythms; harmony is the vertical one of melodies. When we discover the essence of melody composing, we had better consider only itself and disregarding harmony.

Without the theory of harmony, contrapuntal techniques antecedently show the craft to compose and to organize melodies. We have a wealth of laws, rules, customs, and suggestions in modal or tonal counterpoint textbooks (e.g. Jeppensen, 1992 and Kennan, 1987). After Bill Schottstaedt's and Dian-Foon Wu's expert systems (Schottstaedt, 1984 and Wu, 1988), Mary Farbood, Bernd Schoner, Kamil Adiloglu, Ferda N. Alpaslan, and Gabriel Aguilera et al made contributions to the first species counterpoint, too (Farbood, 2001; Adiloglu, 2007; Aguilera, 2010). Beside counterpoint, David Temperley's probabilistic models and David Cope's analyses as well as re-syntheses also pay attention to melody (Temperley, 2007 and Cope, 2001, 2005, and 2008). Even so, none of above went below the note level.

In respect of melody pitches at the note level, there are three layers: the pitch, the interval, and the contour. First, Richard Parncutt explains that musical notes play a role in categorization. "Notes belong to the world of information. The attributes of a note correspond not to the physical attributes of the tone to be played but to its perceptual attributes, expressed by means of labelled categories" (Parncutt, 1989, 23). Such categorization is capable of reducing "the amount of information carried by the pitches of a passage of music to a manageable level, removing information about the precise tuning of pitch or interval, and retaining only its semitone category" (Parncutt, 1989, 44). Second, Kenneth J. Hsu reveals that rather than acoustic frequencies (pitches), “the incidence of the frequency intervals, or of the changes of acoustic frequency, has a fractal geometry” (Hsu, 1991, 3507). Third, not merely Arnold Schoenberg but Leon Dallin mention the importance of a balanced contour to melody composing as well (Schoenberg, 1967 and Dallin, 1974). By contrast with Parncutt, Robert D. Morris claims that “musical contour is one of the most general aspects of pitch perception, prior to the concept of pitch or pitch class” (Morris, 1993, 205). William Thompson also argues that melodic contour is more manageable. “In general, research indicates that listeners’ mental representations of novel melodies contain contour information but relatively little information about absolute pitch or exact interval size” (Thompson, 2008, 95).

On the other hand, we can look those layers through the lower level below notes. Mark Schmuckler surveys models of melodic contour and builds an effective one by Fourier analysis, which can predict the “melodic similarity” (Schmuckler, 2010 and Hewlett, 1998). Nonetheless, its rigid unit of pitch interval (semitones) do not have the capability which Ali C. Gedik has developed, to be “represented in a continuous pitch space in contrast to discrete pitch space representation in western music with 12 pitch-classes” (Gedik, 2010). For this reason, with an eye on a more satisfactory delineation of melodic contour, we could directly count the ratio of audio frequencies instead of the symbolic pitch interval.

Finally, Victor Zuckerkandl describes the effort to cross an interval. “Stepwise motion can be considered normal motion, in the sense that it involves the least effort in the move from tone to tone; while every skip goes beyond the norm in that it expresses a greater effort by taking us to a more distant tone more rapidly than the normal succession of intervening steps would permit” (Zuckerkandl, 1971, 65). In place of the symbolic degree of musical scale, we could measure the effort in ratio of sound energy.

1.3 Purpose

A fairly large body of literature exists on the algorithmic composition of melody. Nevertheless, we cannot apply the predominant symbolic system of pitch class to non-tempered tuning systems which consist of unequal intervals. What is even worse, there is little research has been done on the conception of energy in the motion of melody pitches. Thereby, we need the more universal comprehension with intent to, just as I addressed before, reconsider the phenomena of higher levels from the point of view which we have in lower levels.

In this thesis, I will propose a model for algorithmic composition of melody. More specifically, the melody only comprises monophonic pitches, which melodic contour corresponds with the natural law of kinetic energy and potential energy. Its input and output data could be simply sub-symbolic audio frequencies, but it is also able to accept and produce symbolic musical pitches through translation. After that, I will demonstrate how to utilize the model to implement a system with very limited guidance from music theory.

“Johannes Tinctoris … takes into account the crucial fact that the composer's judgment must be based not only on what he hears at a given moment but what he must keep in mind in the continuity of hearing” (Mann, 1965, viii). By this research, we will have the capability to maintain the continuity of hearing in the motion of melody pitches without statistical processes such as higher-order Markov models, which has “an often overlooked deficiency ... lies in their inability to indicate information which is provided in lower-order models” (Nierhaus, 2009, 81).

Albeit Chih-Fang Huang and I proposed a similar method last year (Lin, 2010), it is not close enough to the natural law of kinetic energy and potential energy. Thereupon I have amended the formulae so that they are simpler yet more logical. I will illustrate the effects in this thesis.

2. Method

2.1 Preliminary

2.1.1 Conservation of Energy

The core concept in this thesis is on the basis the natural law of kinetic energy and potential energy. We can learn about them from physics textbooks. David Halliday, Robert Resnick, and Jearl Walker have written the detail (Halliday, 2005). The equations below (2.1.1~2.1.2) are all from their book.

First of all, the sum of kinetic energy and potential energy present in a mechanical system is the mechanical energy. The following is the equation:

$$\text{“}E_{mec} = K + U\text{”} \qquad (1)$$

- E_{mec}: mechanical energy
- K: kinetic energy of the object
- U: potential energy of the system

Next, “when only conservative forces cause energy transfers within the system—that is, when frictional and drag forces do not act on the objects in the system” (Halliday, 2005, 173), their relationship is in the following way:

$$\text{“}K_2 + U_2 = K_1 + U_1 .\text{”} \qquad (2)$$

It means that the sum of kinetic energy and potential energy in one state is equal to the sum of kinetic energy and potential energy in any other state. To put it differently, “in an isolated system where only conservative forces cause energy changes, the kinetic energy and potential energy can change, but their sum, the mechanical energy E_{mec} of the system, cannot change” (Halliday, 2005, 173). This is the law of conservation of mechanical energy.

Finally, the total energy of a system contains “mechanical energy, thermal energy, and any type of internal energy in addition to thermal energy. ... The total energy of an isolated system cannot change” (Halliday, 2005, 182). This is the law of conservation of energy. Nonetheless, I simply employ the concept of mechanical energy and exclude any other energy in this thesis. Thus, hereafter the conservation mechanical of energy and the conservation of energy are identical.

2.1.2 Average Power of a Wave

Before we continue, I have to clarify three definitions in physics by the following statement:

1. Work: force acting (on an object) through a distance
2. Energy: ability to do work
3. Power: rate of using energy

For instance, first, a force of 10 newtons pushing an object 10 meters in the direction of the force does 100 joules of work. Second, with 100 joules of energy I am able to do 100 joules of work. Third, when I do 100 joules of work in 10 second, my power is 10 watts. Nevertheless, inasmuch as the rate is not germane to the concept in this thesis, we do not need to discriminate between energy and power after this section. Conversely, we should just focus on the transfer of energy (in joule, the common unit which they share).

With the aim of measurement of the effort to cross an interval, we can calculate the ratio of sound energy (power). First of all, we have to know the dependence of the frequency of a wave on the angular frequency. The following is their conversion:

$$\text{“}\,\omega = \frac{2\pi}{T}\,\text{”} \tag{3}$$

$$\text{“}\,f = \frac{1}{T} = \frac{\omega}{2\pi}\,\text{”} \tag{4}$$

- ω: angular frequency
- T: period of oscillation
- f: frequency of a wave

Next, in a transverse wave like the wave on a string, the equation of average power is as follows:

$$\text{“}\,P_{avg} = \frac{1}{2}\mu v \omega^2 {y_m}^2\,\text{”} \tag{5}$$

- P_{avg}: average power
- μ : transverse speed (of the oscillating string element)
- v : wave speed
- ω : angular frequency
- y_m: amplitude

The amplitude “is the magnitude of the maximum displacement of the elements from their equilibrium positions as the wave passes through them. (The subscript m stands for maximum.)” (Halliday, 2005, 416-417) “The factors μ and v in this equation depend on the material and tension of the string. The factors ω and y_m depend on the process that generates the wave” (Halliday, 2005, 424).

Last but not least, in a longitudinal wave such as the sound wave, the theorem is analogous. Its equations are as follows:

$$“I = \frac{P}{A}” \tag{6}$$

$$“I = \frac{1}{2} p v \omega^2 s_m{}^2” \tag{7}$$

- I: intensity of sound
- P: time rate of energy transfer (the power) of the sound wave
- A: area of the surface intercepting the sound
- p: volume density (of air)
- v : wave speed
- ω : angular frequency
- s_m: amplitude

To make a long story short, “the dependence of the average power of a wave on the square of its amplitude and also on the square of its angular frequency is a general result, true for waves of all types” (Halliday, 2005, 424). Namely, if we assume the amplitude is fixed, we will have the ratio of sound power (energy) of two musical pitches by comparing the square of their respective (angular) frequencies.

2.2 Design

2.2.1 Frequency Ratio Interval

Instead of symbolic pitch-classes, we could convert every pitch to its value of audio frequency. Therefore, the musical interval is no longer a generalized symbolic degree of musical scale (see Figure 1). On the contrary, it is the ratio of audio frequencies.

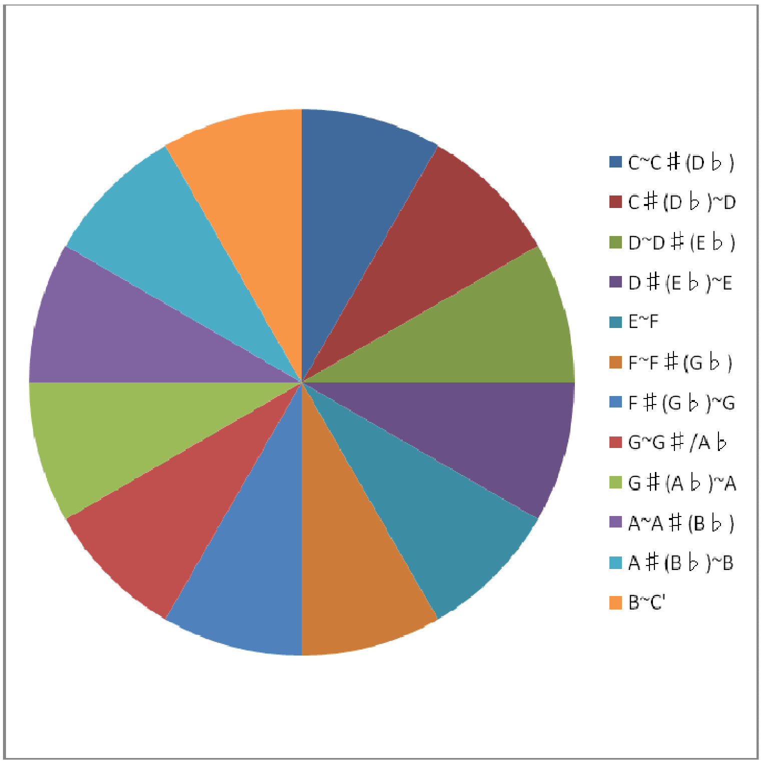


Figure 1. Generalized pitch intervals

When converting musical pitches into audio frequency values, the selection of tuning system is a considerable issue. Talking of western music, although the most common Twelve-tone Equal Temperament with enharmonically equivalent tones is convenient, it might interfere with our thorough appreciation of intervallic quality and tonal harmony (Bobbitt, 1959 and Duffin, 2007). Conversely, the quality and harmony in Just Intonation are most akin to the reality in the performance by virtuosos except instruments in a fixed temperament. Drawing from the frequency ratio intervals on a circle according to Just Intonation (see Table 1 and Table 2), the pictures illustrate that the diatonic pitches spread on uneven positions (see Figure 2 and Figure 3).

Table 1. Frequency ratio intervals on Major Scale

Major Scale (Ionian)	**Do**	**Re**	**Mi**	**Fa**	**Sol**	**La**	**Si**	**Do'**
Given frequency	24	27	30	32	36	40	45	48
Frequency ratio to Tonic	1	9/8	5/4	4/3	3/2	5/3	15/8	2
Distance from Tonic (fractional)	0	1/8	1/4	1/3	1/2	2/3	7/8	1
Distance from Tonic (decimal)	0	0.125	0.25	0.333	0.5	0.666	0.875	1
Distance from Tonic (degree)	0°	45°	90°	120°	180°	240°	315°	360°
Interval (degree)	0°	45°	45°	30°	60°	60°	75°	45°

Table 2. Frequency ratio intervals on Natural Minor Scale

Natural Minor Scale (Aeolian)	**La**	**Si**	**Do**	**Re**	**Mi**	**Fa**	**Sol**	**La'**
Given frequency	120	135	144	160	180	192	216	240
Frequency ratio to Tonic	1	9/8	6/5	4/3	3/2	8/5	9/5	2
Distance from Tonic (fractional)	0	1/8	1/5	1/3	1/2	3/5	4/5	1
Distance from Tonic (decimal)	0	0.125	0.2	0.333	0.5	0.6	0.8	1
Distance from Tonic (degree)	0°	45°	72°	120°	180°	216°	288°	360°
Interval (degree)	0°	45°	27°	48°	60°	36°	72°	72°

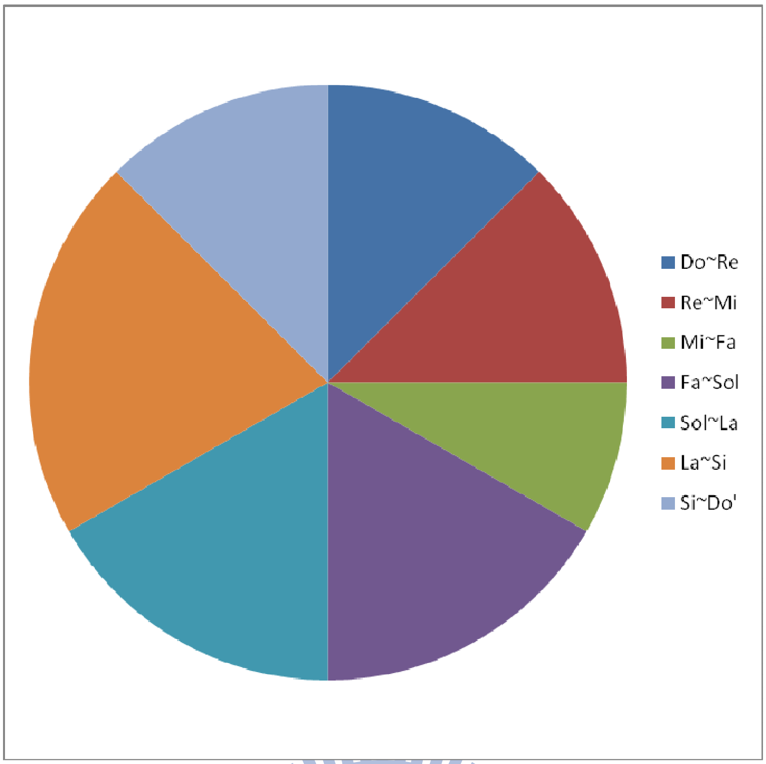


Figure 2. Frequency ratio intervals on Major Scale

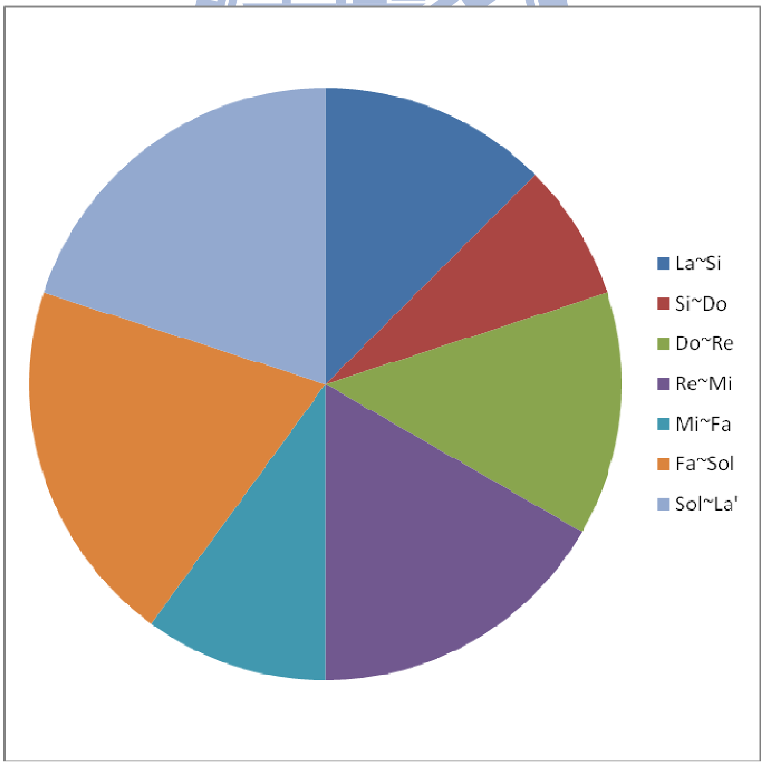


Figure 3. Frequency ratio intervals on Natural Minor Scale

2.2.2 Energy Ratio Interval

When it comes to melody pitches, there are three distinct motion types: ascending movement, descending movement, and standstill. Each of them results in sorts of energy consumption and production. This idea, which is inspired by the gesture of melodic contour, mimics the circumstance of mechanical energy.

Nevertheless, I should emphasize that there is something quite different. Here is a mental exercise: picture two energy containers which provide the energy for ascending movement and descending movement. An ascending movement will lead to costing some energy in its container but infuses it into the container of the descending movement at the same time; an descending movement will lead to costing some energy in its container but infuses it into the container of the ascending movement; static motion simply costs nothing. Therefore, the total energy in both container is always equal. The scenario is just like the law of conservation of energy (2.1.1)

By calculating the ratio of audio frequencies, we could estimate the ratio of sound energy (2.1.2). The difference between two audio frequencies is the energy ratio interval. It also means the energy transfer (consumption and production) between the two energy containers.

2.2.3 Susceptibility

By consulting exhaustive melody rules in textbooks (especially on counterpoint), we may know the first priority is that the melody pitch must return immediately after a large skip. For example, each of the initial four measures below (see Figure 4) is a permissible melody in the third species counterpoint; each of the last two measures (assume that they are all quarter notes) is not. Nonetheless, it usually depends on not only objective situations but also subjective preferences. The fifth measure is acceptable in the second species counterpoint; the last two measures (assume that they are all whole notes) are even tolerable in the first species counterpoint.

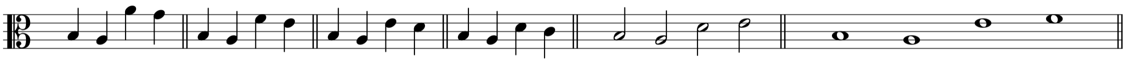



Figure 4. To return or not return after a skip

With the object of discrimination between the varied strictness, I set a parameter named susceptibility. For instance, if the melody pitch (ignore the duration) must return like the fourth measure, its susceptibility is smaller than the melody's in the fifth measure. In other words, it is like a lever which determines the rate of energy transfer (consumption and production) within the melodic contour.

2.2.4 Center Frequency

With the intention of a neutral and natural comparison between the frequency ratios and energy ratios, we must choose a center (pitch) frequency. It is neutral because the chosen frequency is related to neither the focal pitch of melody nor the tonality center of tonal music. On the other hand, ideally speaking, it would be the middle of one's vocal or instrumental comfortable register so that the melody pitches tend to stay within the range. Hence it is also natural.

2.3 Model

The following are the formulae in the light of above opinion:

1. frequency: f, the input audio frequency or the frequency of an input pitch.

2. center frequency: f_0, the given audio frequency as a chosen center pitch.

3. frequency ratio:

$$r_n = \frac{f_n}{f_0}\,; \quad n \geq 0 \tag{8}$$

4. energy ratio:

$$e_n = (r_n)^2\,; \quad n \geq 0 \tag{9}$$

5. energy ratio interval:

$$i_n = e_n - e_{(n-1)}\,; \quad n \geq 1 \tag{10}$$

6. susceptibility constant: *S*, set in advance.

7. initial tolerable energy ratio maximum (always force it ≥ 1): $e\max_1$, set in advance.

8. reciprocal of initial tolerable energy ratio minimum (always force it ≥ 1): $\frac{1}{e\min_1}$, set in advance.

9. tolerable energy ratio maximum (always force it ≥ 1):

$$e\max_n = e\max_{n-1} - (i \times S);\quad n \geq 2 \tag{11}$$

10. reciprocal of tolerable energy ratio minimum (always force it ≥ 1):

$$\frac{1}{e\min_n} = \frac{1}{e\min_{n-1}} + (i \times S);\quad n \geq 2 \tag{12}$$

11. tolerable frequency ratio maximum:

$$r\max_n = \sqrt{e\max_n};\quad n \geq 1 \tag{13}$$

12. reciprocal of tolerable frequency ratio minimum:

$$\frac{1}{r\min_n} = \sqrt{\frac{1}{e\min_n}};\quad n \geq 1 \tag{14}$$

i.e.

$$r\min_n = \sqrt{e\min_n}\ ; \quad n \geq 1 \tag{15}$$

13. normalized frequency ratio:

$$R_n = r_n - 1; \quad n \geq 0 \tag{16}$$

14. normalized tolerable frequency ratio maximum:

$$R\max_n = r\max_n - 1; \quad n \geq 1 \tag{17}$$

15. normalized reciprocal of tolerable frequency ratio minimum:

$$\frac{1}{R\min_n} = -\left(\frac{1}{r\min_{n-1}} - 1\right); \quad n \geq 1 \tag{18}$$

Resulting from the method based on conservation of energy, the sum of “(initial) tolerable energy ratio maximum” and “reciprocal of (initial) tolerable energy ratio minimum” in one state is always equal to it in any other state. Thereupon, the sum of “tolerable frequency ratio maximum” and “reciprocal of tolerable frequency ratio minimum” in one state is always equal to it in any other state, too.

2.4 Proof

“A melody could hardly include unmelodious elements; … The nature and technique of the primordial musical instrument, the voice, determines what is singable. The concept of the melodious in instrumental melody has developed as a free adaptation from the vocal model. (Schoenberg, 1967, 98)” On purpose to prove that the method in this thesis is reasonable, I select the most singable vocal music to test: all the cantus firmus in Aeolian mode and Ionian mode from a textbook (see Figure 5 ~ Figure 13 and Table 3 ~ Table 10) and the widespread Italian art songs (see Figure 14 ~ Figure 17 and Table 11 ~ Table 12). I name the pitches in those art songs by the syllable of

Movable-do Solfège. Additionally, I also choose another song by Schoenberg (Schoenberg, 1967, 110) and give the frequency values according to Twelve-tone Equal Temperament (see Figure 18 ~ Figure 20 and Table 13 ~ Table 14).

In this paragraph, each center (pitch) frequency is roughly inferred from the target melody. In other words, the range of the highest pitch to the lowest pitch in the melody is the comfortable register. Thus the center frequency had better be near the middle of the register. Moreover, the initial tolerable energy ratio maximum (formula 7) and the reciprocal of initial tolerable energy ratio minimum (formula 8) are both always set to "4". It means a probable range between the higher octave and the lower octave. Nevertheless, each of their susceptibilities is various. I maximize it on the basis of the testing results so as to reveal the diverse susceptibilities in the repertoire.

We can observe that the sum of "tolerable energy ratio maximum" and "reciprocal of tolerable energy ratio minimum" in one state is always equal to it in any other state. It is the conservation of energy in those vocal melodic contours.

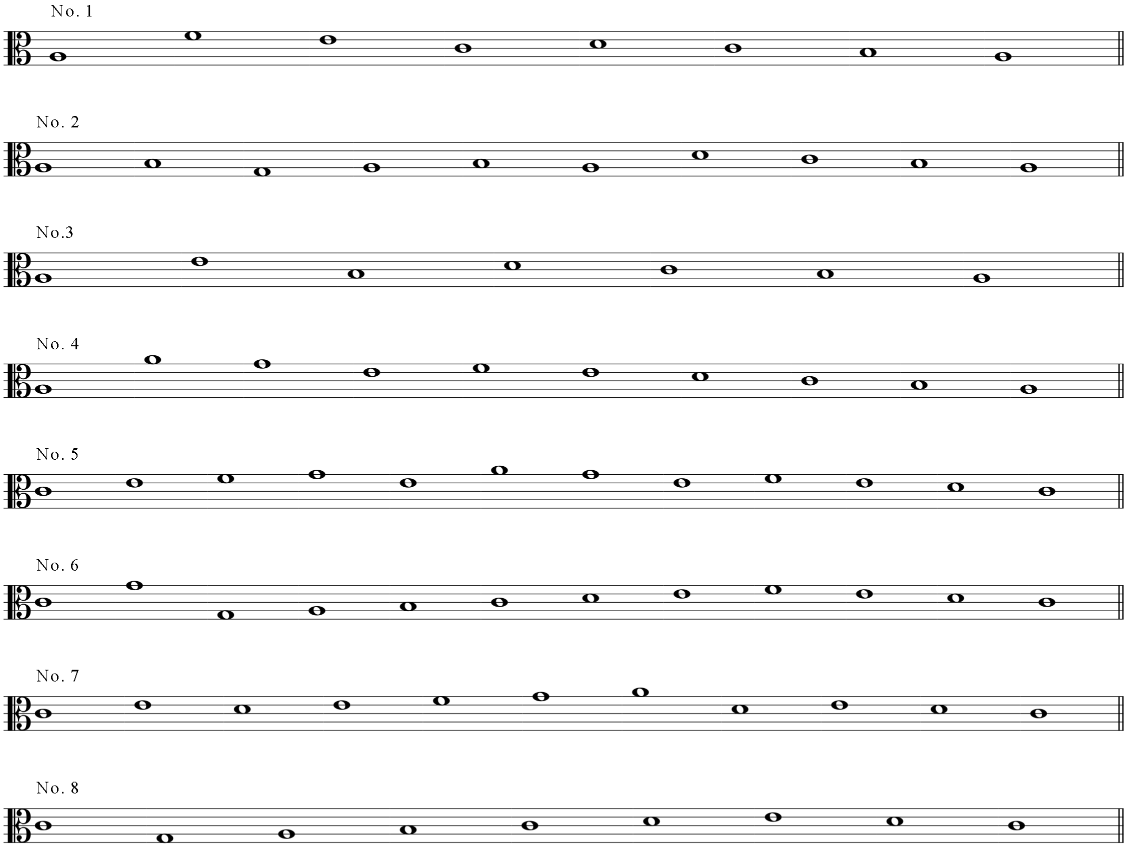


Figure 5. Cantus Firmus (redrawing from Jeppesen, 1992, 108)

Table 3. Analysis of Cantus Firmus No. 1 (S=5)

Pitch	**(re4)**	**la3**	**fa4**	**mi4**	**do4**	**re4**	**do4**	**si3**	**la3**
f	320	240	384	360	288	320	288	270	240
r	1.000	0.750	1.200	1.125	0.900	1.000	0.900	0.844	0.750
e	1.000	0.563	1.440	1.266	0.810	1.000	0.810	0.712	0.563
i		-0.438	0.878	-0.174	-0.456	0.190	-0.190	-0.098	-0.149
e-max		4.000	6.188	1.800	2.672	4.950	4.000	4.950	5.440
1/e-min		4.000	1.813	6.200	5.328	3.050	4.000	3.050	2.560
r-max		2.000	2.487	1.342	1.635	2.225	2.000	2.225	2.332
1/r-min		2.000	1.346	2.490	2.308	1.746	2.000	1.746	1.600
R	0.000	-0.250	0.200	0.125	-0.100	0.000	-0.100	-0.156	-0.250
R-max		1.000	1.487	0.342	0.635	1.225	1.000	1.225	1.332
1/R-min		-1.000	-0.346	-1.490	-1.308	-0.746	-1.000	-0.746	-0.600

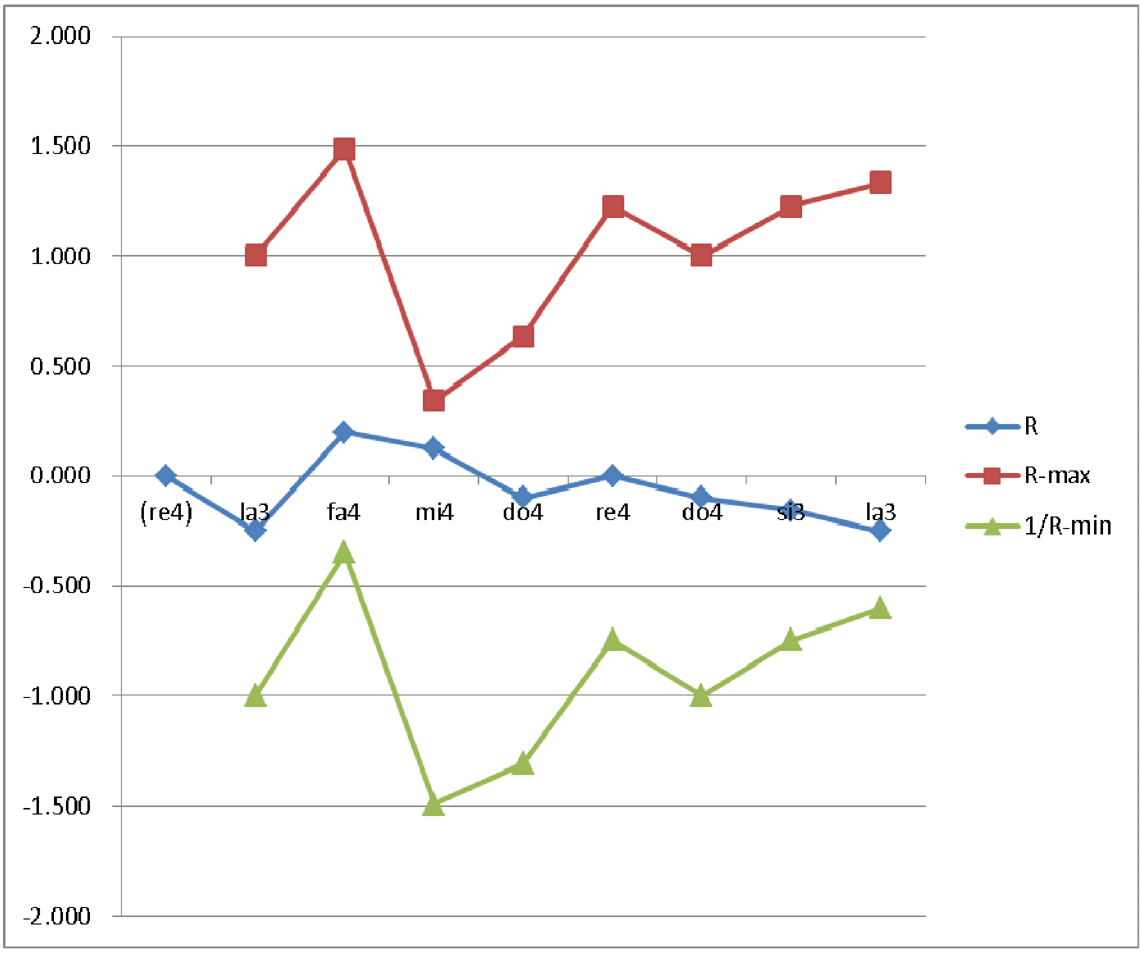


Figure 6. Chart of Table 3

Table 4. Analysis of Cantus Firmus No. 2 (S=6)

Pitch	**(si3)**	**la3**	**si3**	**sol3**	**la3**	**si3**	**la3**	**re4**	**do4**	**si3**	**la3**
f	270	240	270	216	240	270	240	320	288	270	240
r	1.000	0.889	1.000	0.800	0.889	1.000	0.889	1.185	1.067	1.000	0.889
e	1.000	0.790	1.000	0.640	0.790	1.000	0.790	1.405	1.138	1.000	0.790
i		-0.210	0.210	-0.360	0.150	0.210	-0.210	0.615	-0.267	-0.138	-0.210
e-max		4.000	5.259	4.000	6.160	5.259	4.000	5.259	1.572	3.173	4.000
1/e-min		4.000	2.741	4.000	1.840	2.741	4.000	2.741	6.428	4.827	4.000
r-max		2.000	2.293	2.000	2.482	2.293	2.000	2.293	1.254	1.781	2.000
1/r-min		2.000	1.656	2.000	1.356	1.656	2.000	1.656	2.535	2.197	2.000
R	0.000	-0.111	0.000	-0.200	-0.111	0.000	-0.111	0.185	0.067	0.000	-0.111
R-max		1.000	1.293	1.000	1.482	1.293	1.000	1.293	0.254	0.781	1.000
1/R-min		-1.000	-0.656	-1.000	-0.356	-0.656	-1.000	-0.656	-1.535	-1.197	-1.000

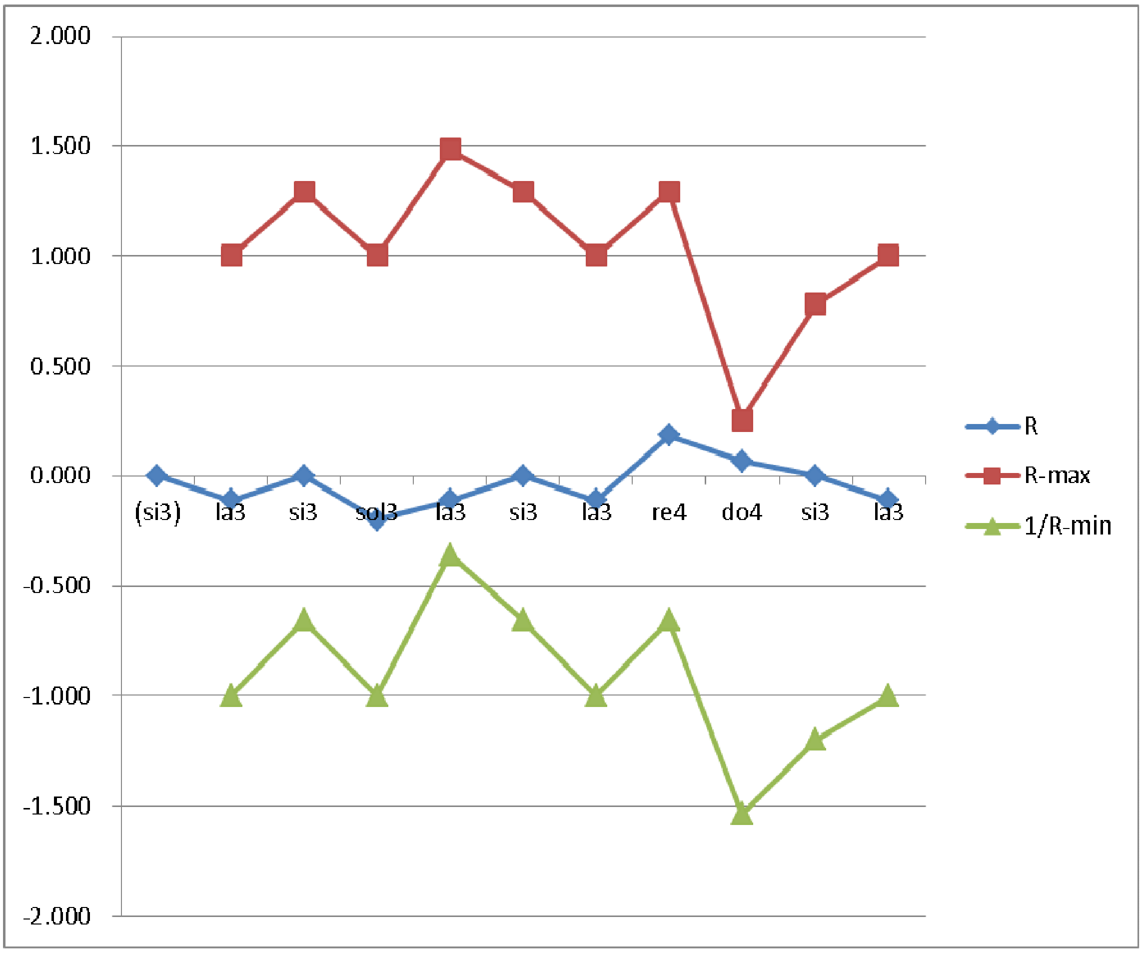


Figure 7. Chart of Table 4

Table 5. Analysis of Cantus Firmus No. 3 (S=5)

Pitch	**(do4)**	**la3**	**mi4**	**si3**	**re4**	**do4**	**si3**	**la3**
f	288	240	360	270	320	288	270	240
r	1.000	0.833	1.250	0.938	1.111	1.000	0.938	0.833
e	1.000	0.694	1.563	0.879	1.235	1.000	0.879	0.694
i		-0.306	0.868	-0.684	0.356	-0.235	-0.121	-0.184
e-max		4.000	5.528	1.188	4.605	2.827	4.000	4.605
1/e-min		4.000	2.472	6.813	3.395	5.173	4.000	3.395
r-max		2.000	2.351	1.090	2.146	1.681	2.000	2.146
1/r-min		2.000	1.572	2.610	1.842	2.274	2.000	1.842
R	0.000	-0.167	0.250	-0.063	0.111	0.000	-0.063	-0.167
R-max		1.000	1.351	0.090	1.146	0.681	1.000	1.146
1/R-min		-1.000	-0.572	-1.610	-0.842	-1.274	-1.000	-0.842

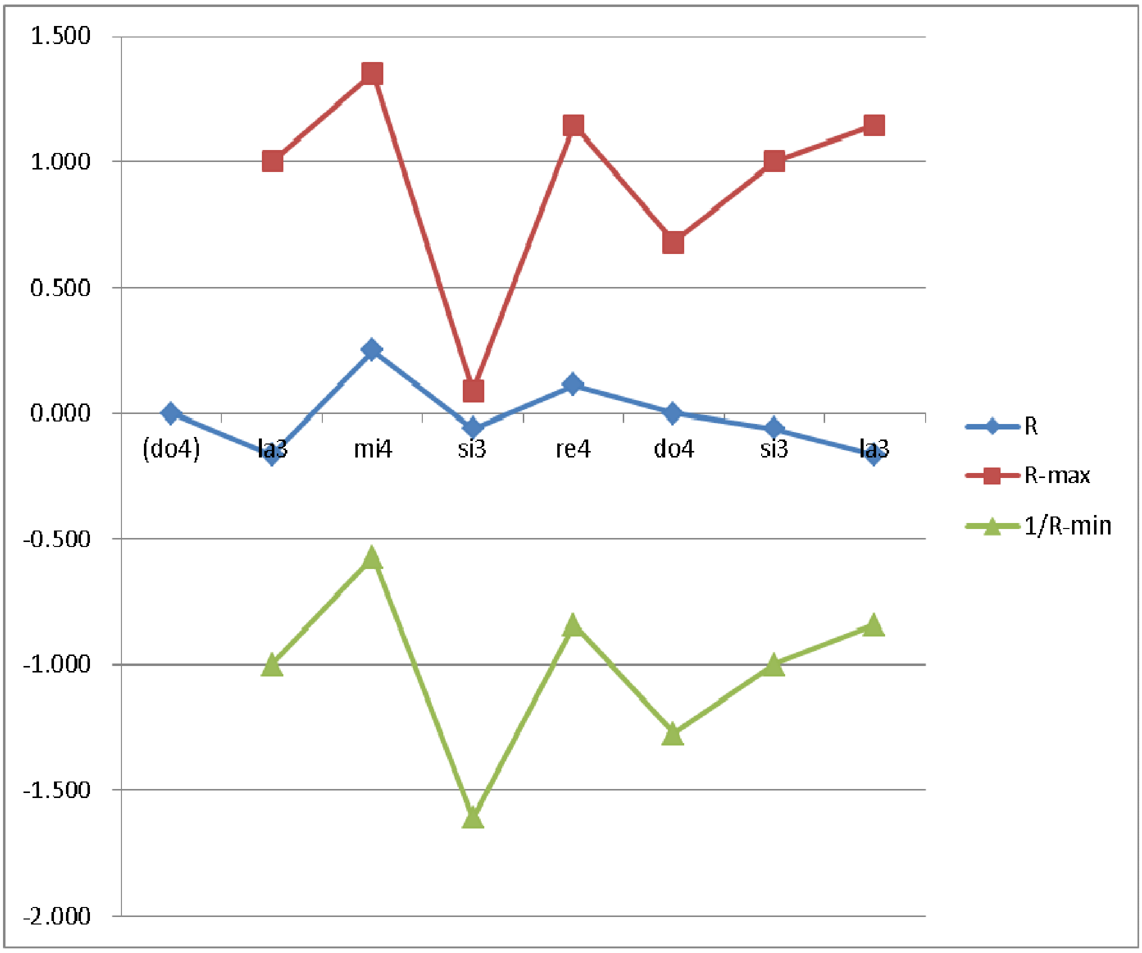


Figure 8. Chart of Table 5

Table 6. Analysis of Cantus Firmus No. 4 (S=3)

Pitch	**(mi4)**	**la3**	**la4**	**sol4**	**mi4**	**fa4**	**mi4**	**re4**	**do4**	**si3**	**la3**
f	360	240	480	432	360	384	360	320	288	270	240
r	1.000	0.667	1.333	1.200	1.000	1.067	1.000	0.889	0.800	0.750	0.667
e	1.000	0.444	1.778	1.440	1.000	1.138	1.000	0.790	0.640	0.563	0.444
i		-0.556	1.333	-0.338	-0.440	0.138	-0.138	-0.210	-0.150	-0.078	-0.118
e-max		4.000	5.667	1.667	2.680	4.000	3.587	4.000	4.630	5.080	5.313
1/e-min		4.000	2.333	6.333	5.320	4.000	4.413	4.000	3.370	2.920	2.688
r-max		2.000	2.380	1.291	1.637	2.000	1.894	2.000	2.152	2.254	2.305
1/r-min		2.000	1.528	2.517	2.307	2.000	2.101	2.000	1.836	1.709	1.639
R	0.000	-0.333	0.333	0.200	0.000	0.067	0.000	-0.111	-0.200	-0.250	-0.333
R-max		1.000	1.380	0.291	0.637	1.000	0.894	1.000	1.152	1.254	1.305
1/R-min		-1.000	-0.528	-1.517	-1.307	-1.000	-1.101	-1.000	-0.836	-0.709	-0.639

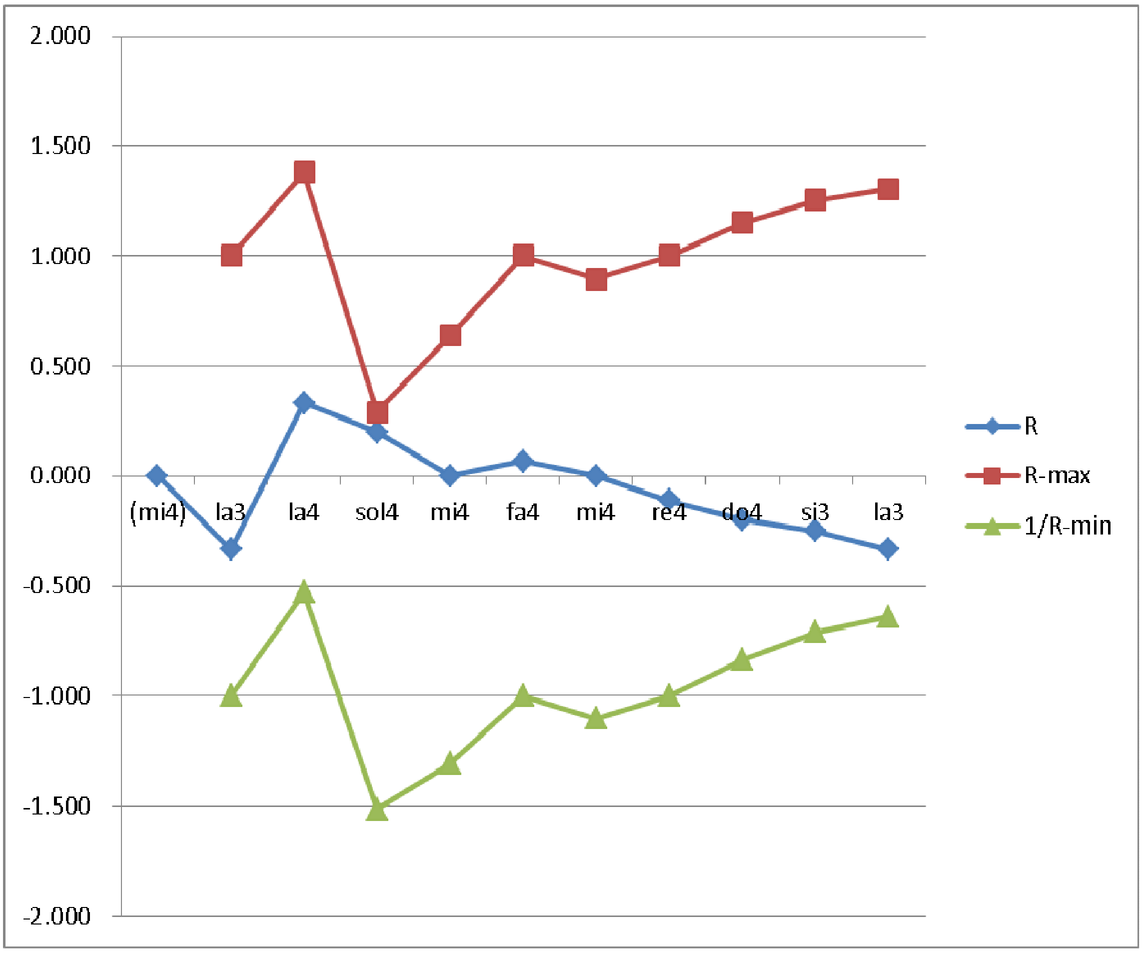


Figure 9. Chart of Table 6

Table 7. Analysis of Cantus Firmus No. 5 (S=3)

Pitch	**(mi4)**	**do4**	**mi4**	**fa4**	**sol4**	**mi4**	**la4**	**sol4**	**mi4**	**fa4**	**mi4**	**re4**	**do4**
f	300	240	300	320	360	300	400	360	300	320	300	270	240
r	1.000	0.800	1.000	1.067	1.200	1.000	1.333	1.200	1.000	1.067	1.000	0.900	0.800
e	1.000	0.640	1.000	1.138	1.440	1.000	1.778	1.440	1.000	1.138	1.000	0.810	0.640
i		-0.360	0.360	0.138	0.302	-0.440	0.778	-0.338	-0.440	0.138	-0.138	-0.190	-0.170
e-max		4.000	5.080	4.000	3.587	2.680	4.000	1.667	2.680	4.000	3.587	4.000	4.570
1/e-min		4.000	2.920	4.000	4.413	5.320	4.000	6.333	5.320	4.000	4.413	4.000	3.430
r-max		2.000	2.254	2.000	1.894	1.637	2.000	1.291	1.637	2.000	1.894	2.000	2.138
1/r-min		2.000	1.709	2.000	2.101	2.307	2.000	2.517	2.307	2.000	2.101	2.000	1.852
R	0.000	-0.200	0.000	0.067	0.200	0.000	0.333	0.200	0.000	0.067	0.000	-0.100	-0.200
R-max		1.000	1.254	1.000	0.894	0.637	1.000	0.291	0.637	1.000	0.894	1.000	1.138
1/R-min		-1.000	-0.709	-1.000	-1.101	-1.307	-1.000	-1.517	-1.307	-1.000	-1.101	-1.000	-0.852

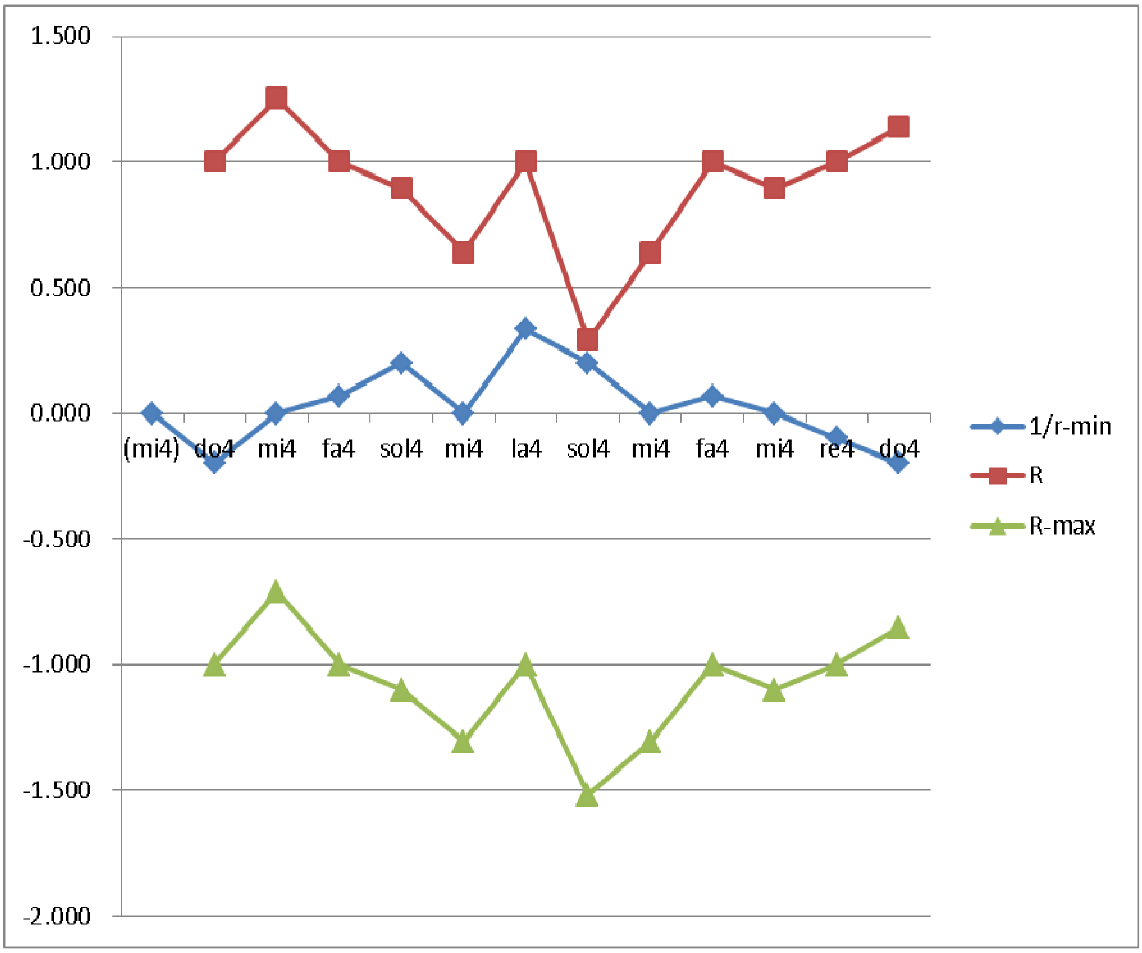


Figure 10. Chart of Table 7

Table 8. Analysis of Cantus Firmus No. 6 (S=4)

Pitch	**(re4)**	**do4**	**sol4**	**sol3**	**la3**	**si3**	**do4**	**re4**	**mi4**	**fa4**	**mi4**	**re4**	**do4**
f	270	240	360	180	200	225	240	270	300	320	300	270	240
r	1.000	0.889	1.333	0.667	0.741	0.833	0.889	1.000	1.111	1.185	1.111	1.000	0.889
e	1.000	0.790	1.778	0.444	0.549	0.694	0.790	1.000	1.235	1.405	1.235	1.000	0.790
i		-0.210	0.988	-1.333	0.104	0.146	0.096	0.210	0.235	0.170	-0.170	-0.235	-0.210
e-max		4.000	4.840	0.889	6.222	5.805	5.222	4.840	4.000	3.062	2.381	3.062	4.000
1/e-min		4.000	3.160	7.111	1.778	2.195	2.778	3.160	4.000	4.938	5.619	4.938	4.000
r-max		2.000	2.200	0.943	2.494	2.409	2.285	2.200	2.000	1.750	1.543	1.750	2.000
1/r-min		2.000	1.778	2.667	1.333	1.481	1.667	1.778	2.000	2.222	2.370	2.222	2.000
R	0.000	-0.111	0.333	-0.333	-0.259	-0.167	-0.111	0.000	0.111	0.185	0.111	0.000	-0.111
R-max		1.000	1.200	-0.057	1.494	1.409	1.285	1.200	1.000	0.750	0.543	0.750	1.000
1/R-min		-1.000	-0.778	-1.667	-0.333	-0.481	-0.667	-0.778	-1.000	-1.222	-1.370	-1.222	-1.000

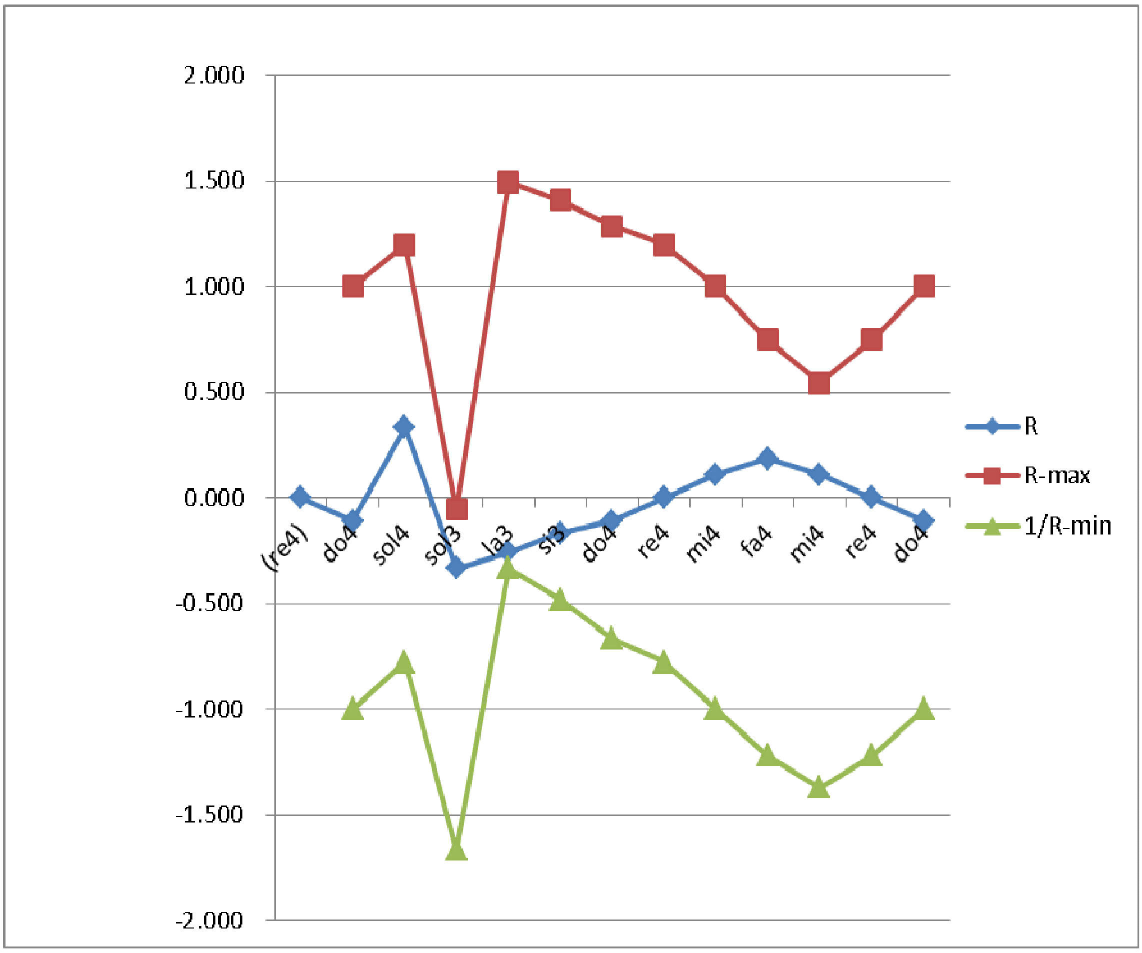


Figure 11. Chart of Table 8

Table 9. Analysis of Cantus Firmus No. 7 (S=4)

Pitch	**(mi4)**	**do4**	**mi4**	**re4**	**mi4**	**fa4**	**sol4**	**la4**	**re4**	**mi4**	**re4**	**do4**
f	300	240	300	270	300	320	360	400	270	300	270	240
r	1.000	0.800	1.000	0.900	1.000	1.067	1.200	1.333	0.900	1.000	0.900	0.800
e	1.000	0.640	1.000	0.810	1.000	1.138	1.440	1.778	0.810	1.000	0.810	0.640
i		-0.360	0.360	-0.190	0.190	0.138	0.302	0.338	-0.968	0.190	-0.190	-0.170
e-max		4.000	5.440	4.000	4.760	4.000	3.449	2.240	0.889	4.760	4.000	4.760
1/e-min		4.000	2.560	4.000	3.240	4.000	4.551	5.760	7.111	3.240	4.000	3.240
r-max		2.000	2.332	2.000	2.182	2.000	1.857	1.497	0.943	2.182	2.000	2.182
1/r-min		2.000	1.600	2.000	1.800	2.000	2.133	2.400	2.667	1.800	2.000	1.800
R	0.000	-0.200	0.000	-0.100	0.000	0.067	0.200	0.333	-0.100	0.000	-0.100	-0.200
R-max		1.000	1.332	1.000	1.182	1.000	0.857	0.497	-0.057	1.182	1.000	1.182
1/R-min		-1.000	-0.600	-1.000	-0.800	-1.000	-1.133	-1.400	-1.667	-0.800	-1.000	-0.800

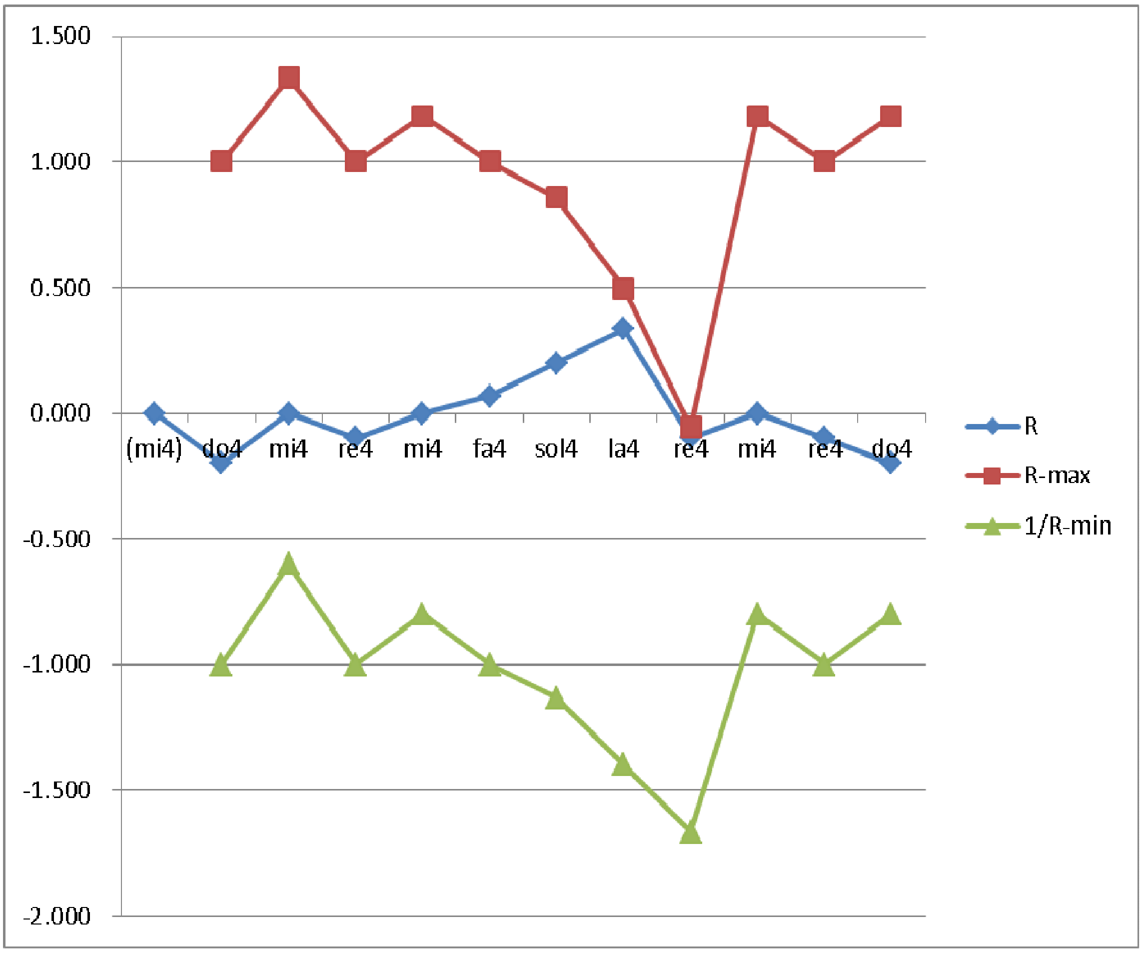


Figure 12. Chart of Table 9

Table 10. Analysis of Cantus Firmus No. 8 (S=4)

Pitch	**(do4)**	**do4**	**sol3**	**la3**	**si3**	**do4**	**re4**	**mi4**	**re4**	**do4**
f	240	240	180	200	225	240	270	300	270	240
r	1.000	1.000	0.750	0.833	0.938	1.000	1.125	1.250	1.125	1.000
e	1.000	1.000	0.563	0.694	0.879	1.000	1.266	1.563	1.266	1.000
i		0.000	-0.438	0.132	0.184	0.121	0.266	0.297	-0.297	-0.266
e-max		4.000	4.000	5.750	5.222	4.484	4.000	2.938	1.750	2.938
1/e-min		4.000	4.000	2.250	2.778	3.516	4.000	5.063	6.250	5.063
r-max		2.000	2.000	2.398	2.285	2.118	2.000	1.714	1.323	1.714
1/r-min		2.000	2.000	1.500	1.667	1.875	2.000	2.250	2.500	2.250
R	0.000	0.000	-0.250	-0.167	-0.063	0.000	0.125	0.250	0.125	0.000
R-max		1.000	1.000	1.398	1.285	1.118	1.000	0.714	0.323	0.714
1/R-min		-1.000	-1.000	-0.500	-0.667	-0.875	-1.000	-1.250	-1.500	-1.250

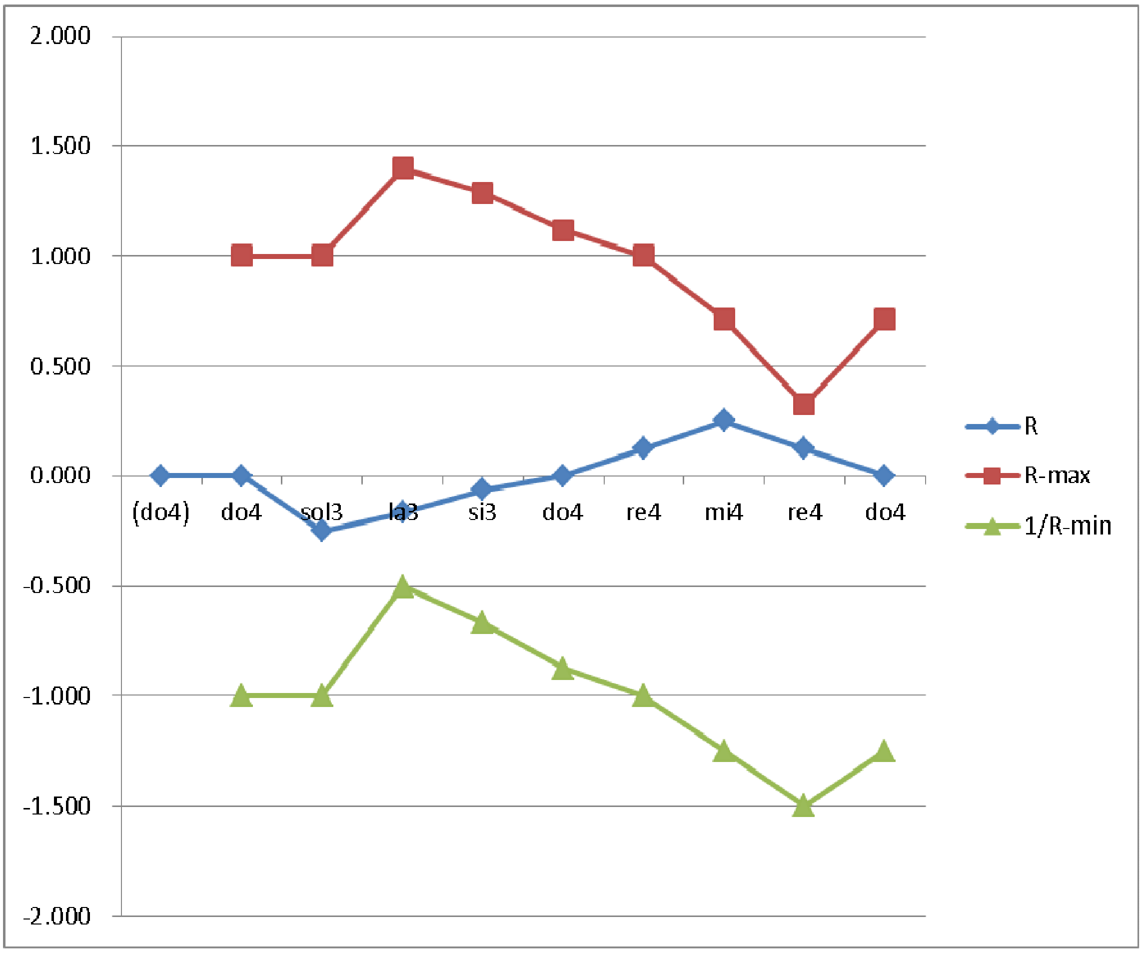


Figure 13. Chart of Table 10

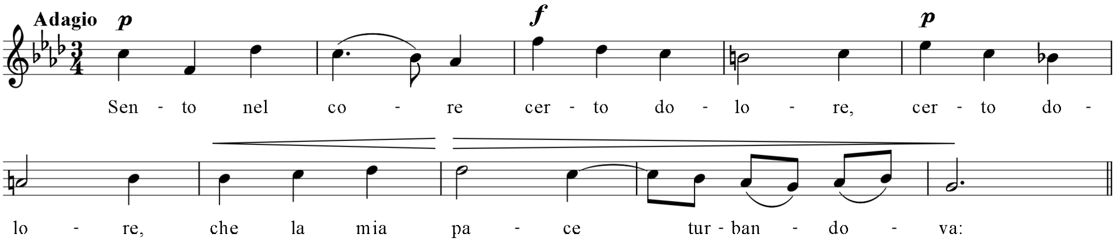


Figure 14. A. Scarlatti (1659~1725): "Sento nel core", mm. 5~14

Table 11. Analysis of Figure 14 (S=3)

Pitch	**(mi4)**	**mi4**	**la3**	**fa4**	**mi4**	**re4**	**do4**	**la4**	**fa4**	**mi4**	**ri4**	**mi4**	**sol4**	**mi4**	**re4**	**di4**	**re4**
f	360	360	240	384	360	320	288	480	384	360	338	360	432	360	320	300	320
r	1.000	1.000	0.667	1.067	1.000	0.889	0.800	1.333	1.067	1.000	0.938	1.000	1.200	1.000	0.889	0.833	0.889
e	1.000	1.000	0.444	1.138	1.000	0.790	0.640	1.778	1.138	1.000	0.879	1.000	1.440	1.000	0.790	0.694	0.790
i		0.000	-0.556	0.693	-0.138	-0.210	-0.150	1.138	-0.640	-0.138	-0.121	0.121	0.440	-0.440	-0.210	-0.096	0.096
e-max		4.000	4.000	5.667	3.587	4.000	4.630	5.080	1.667	3.587	4.000	4.363	4.000	2.680	4.000	4.630	4.917
1/e-min		4.000	4.000	2.333	4.413	4.000	3.370	2.920	6.333	4.413	4.000	3.637	4.000	5.320	4.000	3.370	3.083
r-max		2.000	2.000	2.380	1.894	2.000	2.152	2.254	1.291	1.894	2.000	2.089	2.000	1.637	2.000	2.152	2.217
1/r-min		2.000	2.000	1.528	2.101	2.000	1.836	1.709	2.517	2.101	2.000	1.907	2.000	2.307	2.000	1.836	1.756
R	0.000	0.000	-0.333	0.067	0.000	-0.111	-0.200	0.333	0.067	0.000	-0.063	0.000	0.200	0.000	-0.111	-0.167	-0.111
R-max		1.000	1.000	1.380	0.894	1.000	1.152	1.254	0.291	0.894	1.000	1.089	1.000	0.637	1.000	1.152	1.217
1/R-min		-1.000	-1.000	-0.528	-1.101	-1.000	-0.836	-0.709	-1.517	-1.101	-1.000	-0.907	-1.000	-1.307	-1.000	-0.836	-0.756

(Table 11 continued)

Pitch	**re4**	**mi4**	**fa4**	**fa4**	**mi4**	**re4**	**do4**	**si3**	**do4**	**re4**	**si3**
f	320	360	384	384	360	320	288	270	288	320	270
r	0.889	1.000	1.067	1.067	1.000	0.889	0.800	0.750	0.800	0.889	0.750
e	0.790	1.000	1.138	1.138	1.000	0.790	0.640	0.563	0.640	0.790	0.563
i	0.000	0.210	0.138	0.000	-0.138	-0.210	-0.150	-0.078	0.078	0.150	-0.228
e-max	4.630	4.630	4.000	3.587	3.587	4.000	4.630	5.080	5.313	5.080	4.630
1/e-min	3.370	3.370	4.000	4.413	4.413	4.000	3.370	2.920	2.688	2.920	3.370
r-max	2.152	2.152	2.000	1.894	1.894	2.000	2.152	2.254	2.305	2.254	2.152
1/r-min	1.836	1.836	2.000	2.101	2.101	2.000	1.836	1.709	1.639	1.709	1.836
R	-0.111	0.000	0.067	0.067	0.000	-0.111	-0.200	-0.250	-0.200	-0.111	-0.250
R-max	1.152	1.152	1.000	0.894	0.894	1.000	1.152	1.254	1.305	1.254	1.152
1/R-min	-0.836	-0.836	-1.000	-1.101	-1.101	-1.000	-0.836	-0.709	-0.639	-0.709	-0.836

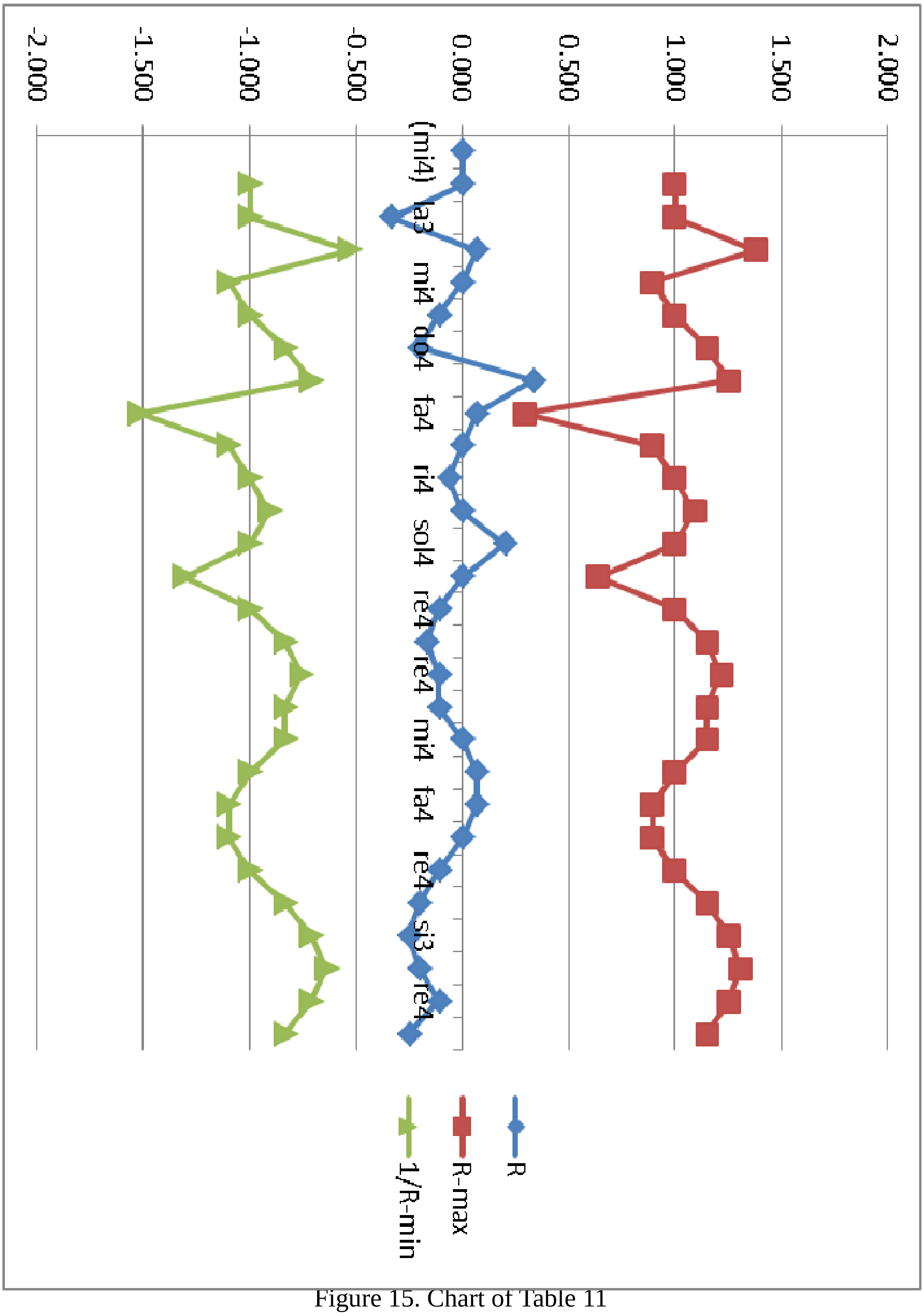


Figure 15. Chart of Table 11

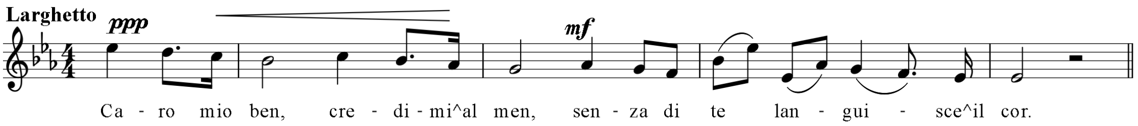


Figure 16. G. Giordani (1743~1798): "Caro mio ben", mm. 22~26

Table 12. Analysis of Figure 16 (S=3)

Pitch	**(sol4)**	**do5**	**si4**	**la4**	**sol4**	**la4**	**sol4**	**fa4**	**mi4**
f	360	480	450	400	360	400	360	320	300
r	1.000	1.333	1.250	1.111	1.000	1.111	1.000	0.889	0.833
e	1.000	1.778	1.563	1.235	1.000	1.235	1.000	0.790	0.694
i		0.778	-0.215	-0.328	-0.235	0.235	-0.235	-0.210	-0.096
e-max		4.000	1.667	2.313	3.296	4.000	3.296	4.000	4.630
1/e-min		4.000	6.333	5.688	4.704	4.000	4.704	4.000	3.370
r-max		2.000	1.291	1.521	1.816	2.000	1.816	2.000	2.152
1/r-min		2.000	2.517	2.385	2.169	2.000	2.169	2.000	1.836
R	0.000	0.333	0.250	0.111	0.000	0.111	0.000	-0.111	-0.167
R-max		1.000	0.291	0.521	0.816	1.000	0.816	1.000	1.152
1/R-min		-1.000	-1.517	-1.385	-1.169	-1.000	-1.169	-1.000	-0.836

(Table 12 continued)

Pitch	**fa4**	**mi4**	**re4**	**sol4**	**do5**	**do4**	**fa4**	**mi4**	**re4**	**do4**	**do4**
f	320	300	270	360	480	240	320	300	270	240	240
r	0.889	0.833	0.750	1.000	1.333	0.667	0.889	0.833	0.750	0.667	0.667
e	0.790	0.694	0.563	1.000	1.778	0.444	0.790	0.694	0.563	0.444	0.444
i	0.096	-0.096	-0.132	0.438	0.778	-1.333	0.346	-0.096	-0.132	-0.118	0.000
e-max	4.917	4.630	4.917	5.313	4.000	1.667	5.667	4.630	4.917	5.313	5.667
1/e-min	3.083	3.370	3.083	2.688	4.000	6.333	2.333	3.370	3.083	2.688	2.333
r-max	2.217	2.152	2.217	2.305	2.000	1.291	2.380	2.152	2.217	2.305	2.380
1/r-min	1.756	1.836	1.756	1.639	2.000	2.517	1.528	1.836	1.756	1.639	1.528
R	-0.111	-0.167	-0.250	0.000	0.333	-0.333	-0.111	-0.167	-0.250	-0.333	-0.333
R-max	1.217	1.152	1.217	1.305	1.000	0.291	1.380	1.152	1.217	1.305	1.380
1/R-min	-0.756	-0.836	-0.756	-0.639	-1.000	-1.517	-0.528	-0.836	-0.756	-0.639	-0.528

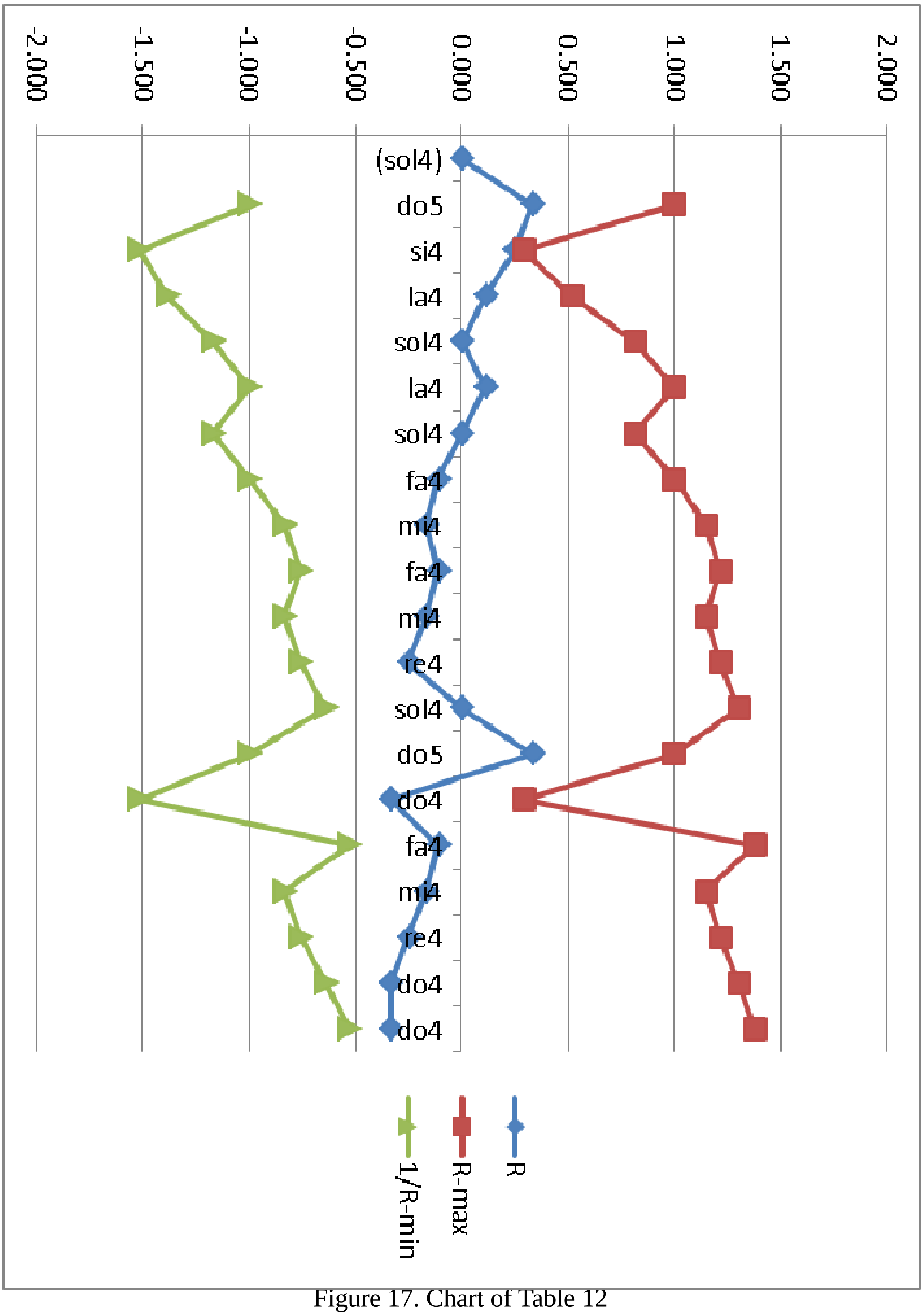


Figure 17. Chart of Table 12

Schoenberg: 8 Lieder, Op. 6 1

Schoenberg
Traumleben
(Hart)
Op. 6, No. 1

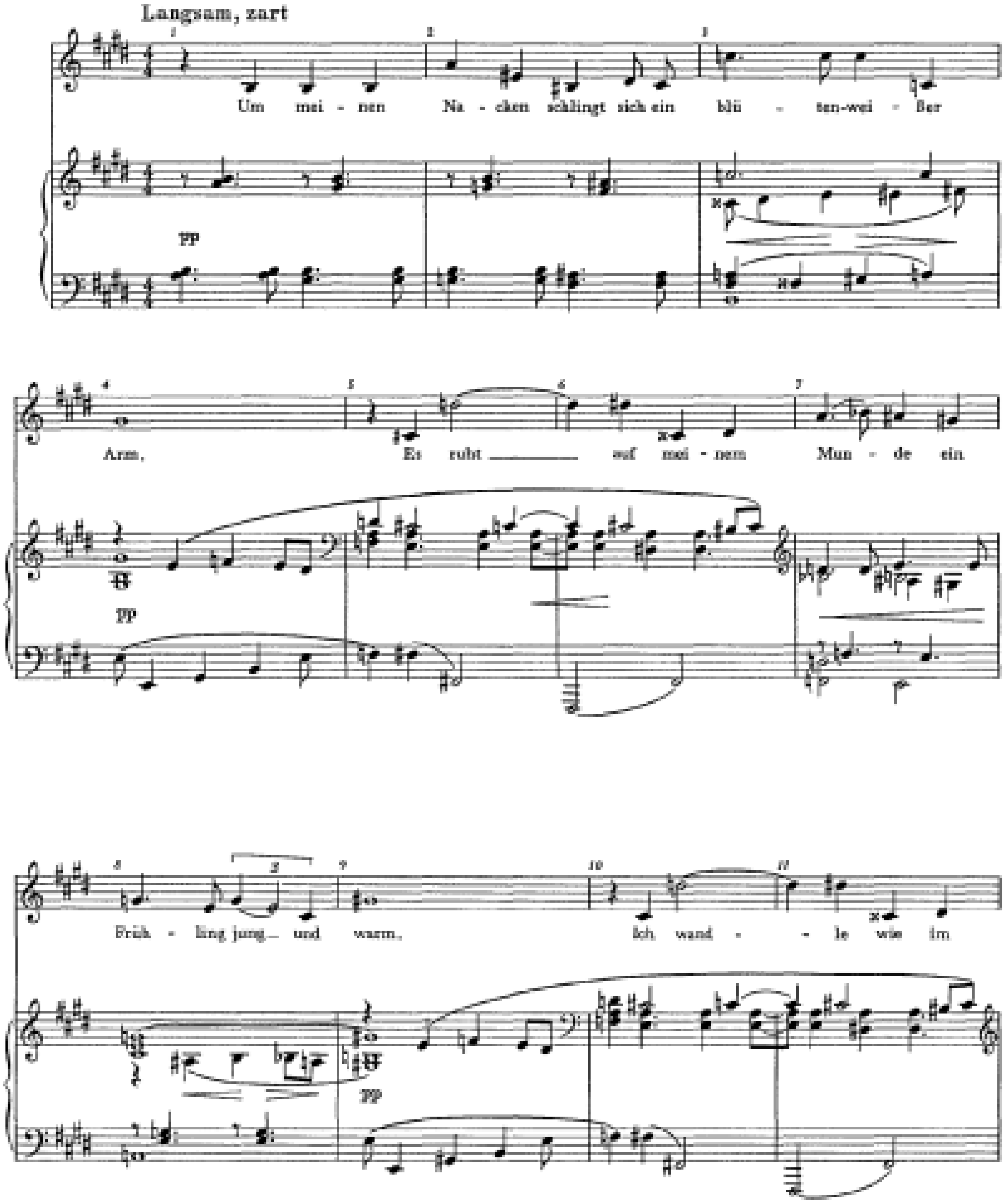



Figure 18. A. Schoenberg (1874~1915): Op. 6, No. 1, mm. 1~11 (Schoenberg, 1908)

Table 13. Analysis of the voice in mm. 1~4 of Figure 18 (S=2.5)

Pitch	**(G4)**	**B3**	**B3**	**B3**	**A4**	**E4♯**	**B3♯**	**D4♯**	**C4♯**	**C5**	**C5**	**C5**	**C4**	**G4♯**
f	392.00	246.94	246.94	246.94	440.00	349.23	261.60	311.13	277.18	523.25	523.25	523.25	261.60	415.30
r	1.000	0.630	0.630	0.630	1.122	0.891	0.667	0.794	0.707	1.335	1.335	1.335	0.667	1.059
e	1.000	0.397	0.397	0.397	1.260	0.794	0.445	0.630	0.500	1.782	1.782	1.782	0.445	1.122
i		-0.603	0.000	0.000	0.863	-0.466	-0.348	0.185	-0.130	1.282	0.000	0.000	-1.336	0.677
e-max		4.000	5.508	5.508	5.508	3.350	4.516	5.387	4.925	5.250	2.046	2.046	2.046	5.387
1/e-min		4.000	2.492	2.492	2.492	4.650	3.484	2.613	3.075	2.750	5.954	5.954	5.954	2.613
r-max		2.000	2.347	2.347	2.347	1.830	2.125	2.321	2.219	2.291	1.430	1.430	1.430	2.321
1/r-min		2.000	1.579	1.579	1.579	2.156	1.867	1.617	1.754	1.658	2.440	2.440	2.440	1.617
R	0.000	-0.370	-0.370	-0.370	0.122	-0.109	-0.333	-0.206	-0.293	0.335	0.335	0.335	-0.333	0.059
R-max		1.000	1.347	1.347	1.347	0.830	1.125	1.321	1.219	1.291	0.430	0.430	0.430	1.321
1/R-min		-1.000	-0.579	-0.579	-0.579	-1.156	-0.867	-0.617	-0.754	-0.658	-1.440	-1.440	-1.440	-0.617

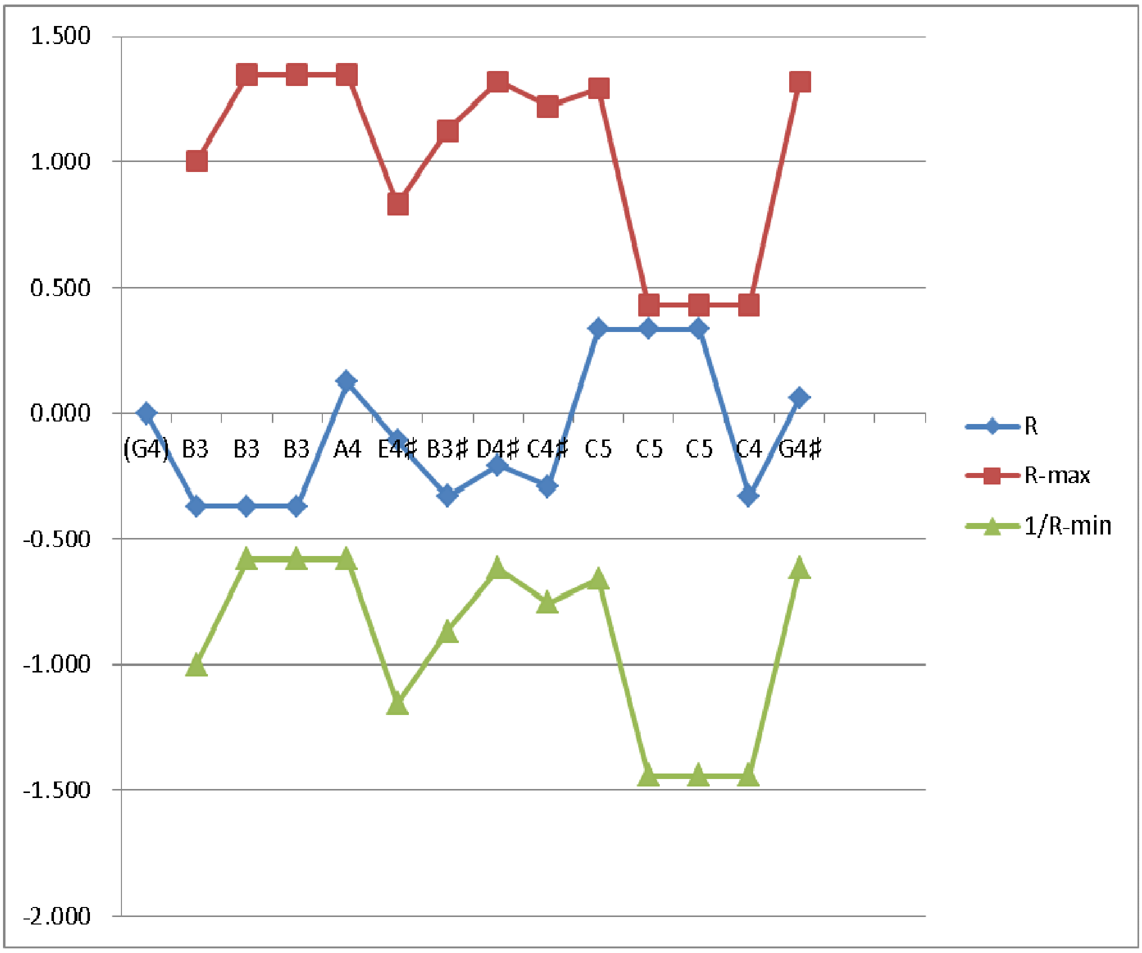


Figure 19. Chart of Table 13

Table 14. Analysis of the voice in mm. 5~9 of Figure 18 (S=1)

Pitch	(G4)	C4＃	D5	D5＃	C4＃＃	D4＃	A4	B4♭	A4＃	G4＃	G4	E4	G4	E4	C4＃	G4＃
f	392.0	277.2	587.3	622.3	293.7	311.1	440.0	466.2	466.2	415.3	392.0	329.6	392.0	329.6	277.2	415.3
r	1.000	0.707	1.498	1.587	0.749	0.794	1.122	1.189	1.189	1.059	1.000	0.841	1.000	0.841	0.707	1.059
e	1.000	0.500	2.245	2.520	0.561	0.630	1.260	1.414	1.414	1.122	1.000	0.707	1.000	0.707	0.500	1.122
i		-0.500	1.745	0.275	-1.959	0.069	0.630	0.154	0.000	-0.292	-0.122	-0.293	0.293	-0.293	-0.207	0.622
e-max		4.000	4.500	2.755	2.480	4.439	4.370	3.740	3.586	3.586	3.878	4.000	4.293	4.000	4.293	4.500
1/e-min		4.000	3.500	5.245	5.520	3.561	3.630	4.260	4.414	4.414	4.122	4.000	3.707	4.000	3.707	3.500
r-max		2.000	2.121	1.660	1.575	2.107	2.090	1.934	1.894	1.894	1.969	2.000	2.072	2.000	2.072	2.121
1/r-min		2.000	1.871	2.290	2.349	1.887	1.905	2.064	2.101	2.101	2.030	2.000	1.925	2.000	1.925	1.871
R	0.000	-0.293	0.498	0.587	-0.251	-0.206	0.122	0.189	0.189	0.059	0.000	-0.159	0.000	-0.159	-0.293	0.059
R-max		1.000	1.121	0.660	0.575	1.107	1.090	0.934	0.894	0.894	0.969	1.000	1.072	1.000	1.072	1.121
1/R-min		-1.000	-0.871	-1.290	-1.349	-0.887	-0.905	-1.064	-1.101	-1.101	-1.030	-1.000	-0.925	-1.000	-0.925	-0.871

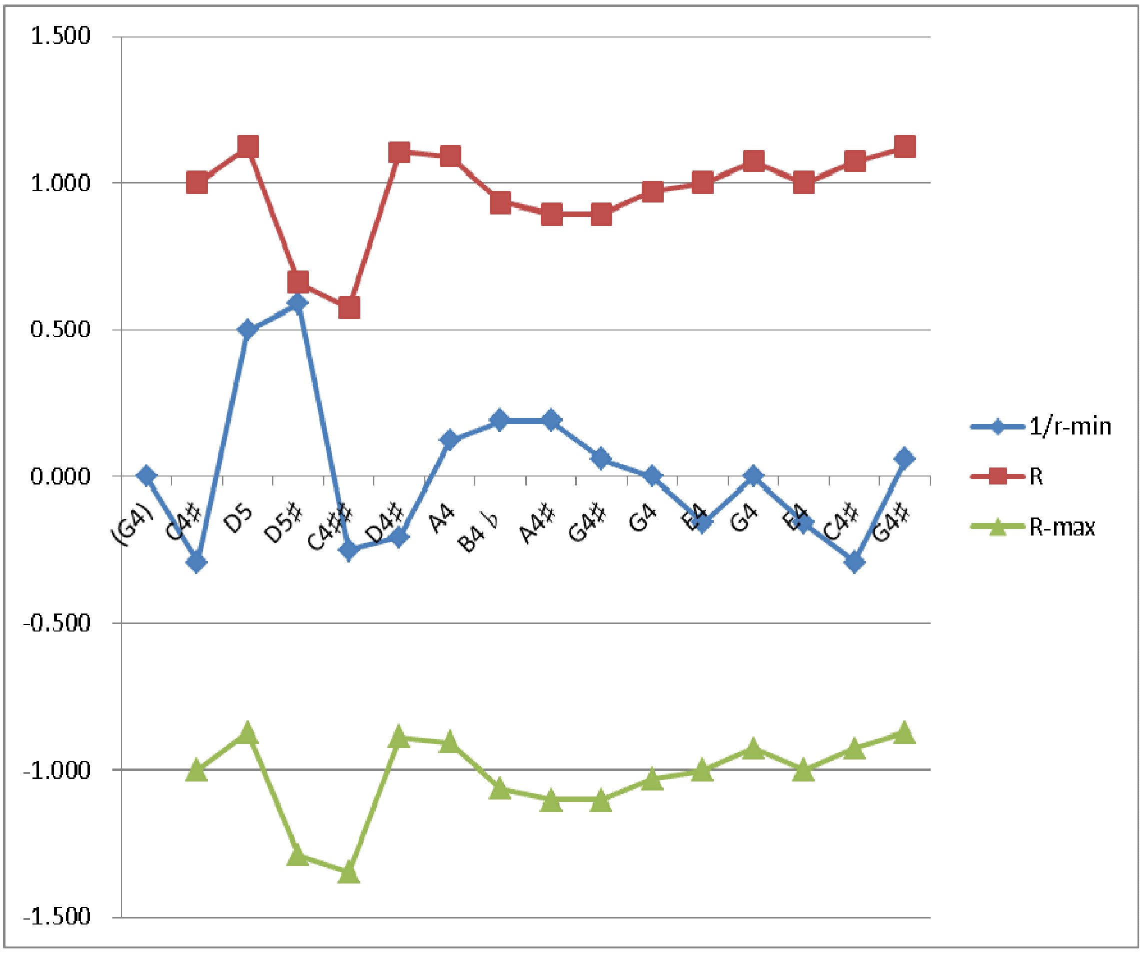


Figure 20. Chart of Table 14

2.5 Restrictions

Although the model is basically valid as above, there is a noticeable hazard. If we use it to compose melody pitches, the melodic contour will have a certain chance to be excessively static or dynamic. In the former condition, there will be too many steps and repetitions but too few skips. It is safe to use an abundance of steps and repetitions on purpose to prevent from violating melody composing rules, whereas the product may be ridiculously boring. By contrast, there will be too many skips in the latter condition. It may still correspond with the model while the incredible discrete frequencies will be perceived as pitches from voices rather than a single melody part.

In answer to circumventing the risk, we need several constraints on the motion of melody pitches. They are like as follows:

1. maximal times of consecutive step in the same direction or in similar directions
2. maximal times of consecutive step in contrary directions
3. maximal times of consecutive step, which is not fewer than each of above two
4. maximal times of consecutive skip
5. maximal times of pitch repetition

In terms of the first constraint, if three successive pitches are all ascending or descending, they are in the same direction; if one of them is a repetition, they are in similar directions.

Consecutive skips, whether in the same or contrary directions, must be used cautiously when we compose a melody. Furthermore, the consecutive skips in the same direction are already under control on account of the conservation of energy in the model. Therefore, I only set the maximal times of consecutive skip and disregard the direction.

When the input data are audio frequencies instead of musical pitches, we can just set a threshold for recognizing the ratio of successive frequencies. If the ratio is greater than the threshold, it will be defined as a skip. Hence, by adding those restrictions, we could expect a sensible melodic contour and feasible melody motions.

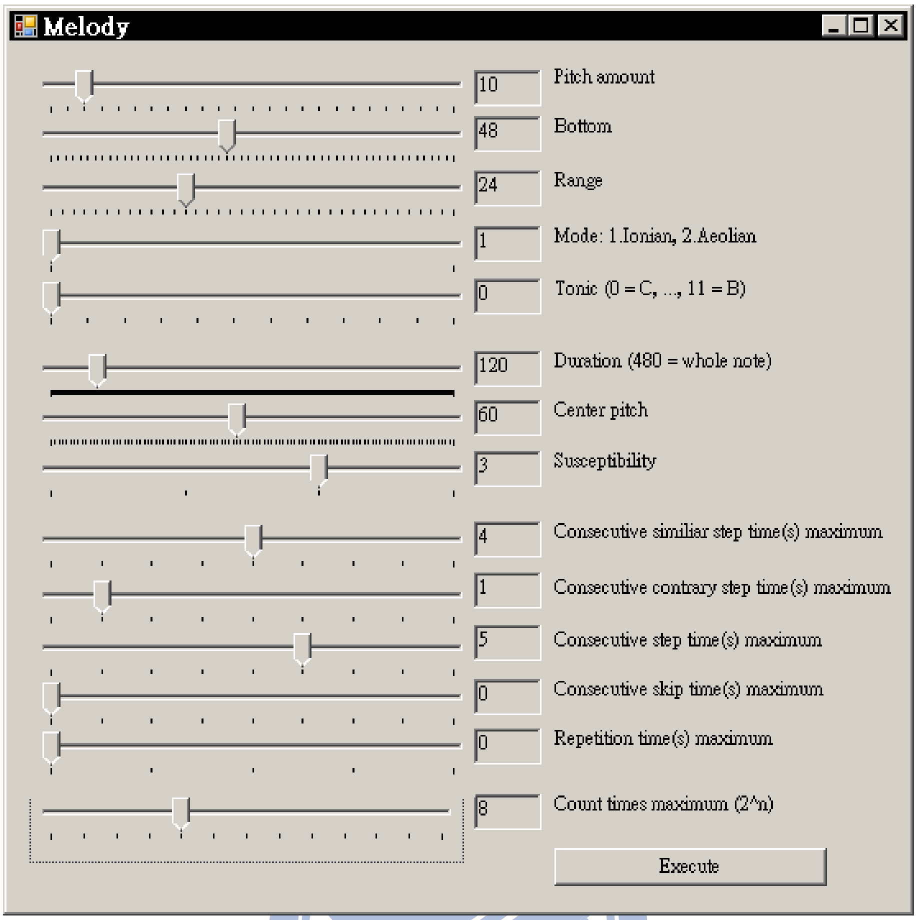
3. Implementation

3.1 Structure

3.1.1 Parameters

On purpose to simplify the procedure and make the output result be more understandable, we had better still rely upon the symbolic notation to present. I build a small program with GUI (graphic user interface) (see Figure 21), which is written in Microsoft Visual C++ 2010 Express Edition using Windows API (Application Programming Interfaces) and .NET Framework 4. The interface provides several style parameters for restriction and preference. They are respectively in the following way:

1. total amount of output notes
2. the lowest possible pitch of melody
3. range above the lowest possible pitch, for the determination of the register
4. musical scale: Ionian mode or Aeolian mode
5. tonic pitch
6. duration of each pitch
7. center pitch, for resolving f_0
8. susceptibility, for the determination of S.
9. maximal times of consecutive step in the same direction or in similar directions
10. maximal times of consecutive step in contrary directions
11. maximal times of consecutive step, which is always greater than each of above two
12. maximal times of consecutive skip
13. maximal times of pitch repetition
14. maximal count times of the computer program, for the backtracking mechanism to prevent from infinite loops



10 Pitch amount
48 Bottom
24 Range
1 Mode: 1.Ionian, 2.Aeolian
0 Tonic (0 = C, ..., 11 = B)
120 Duration (480 = whole note)
60 Center pitch
3 Susceptibility
4 Consecutive similiar step time(s) maximum
1 Consecutive contrary step time(s) maximum
5 Consecutive step time(s) maximum
0 Consecutive skip time(s) maximum
0 Repetition time(s) maximum
8 Count times maximum (2^n)

Figure 21. Program screenshot

3.1.2 Algorithm

The whole system structure is uncomplicated (see Figure 22). Some small routines are on the basis of Phil Winsor's instructions (Winsor, 1989 and 1991). In this chapter, both the initial tolerable energy ratio maximum (formula 7) and the reciprocal of initial tolerable energy ratio minimum (formula 8) are also always set to "4" (as chapter two).

The first chosen pitch bypasses both the "acceptable interval" and the "melody motion" routine because merely one pitch cannot indicate any information about them. Furthermore, we need successive three pitches to identify the type of consecutive melody motion. Hence the second chosen pitch bypasses the "melody motion" routine, too.

The final output file is simply a plain text file. It has a special content which is for the Score File (*.sc) of MusicSculptor, a 16-bit software (see Figure 23 and Figure 24) by Winsor and Kuo-Lung Chang. It can only runs on the DOS (Disk Operating System) platform. The Score File can be easily converted into Standard Midi File (*.mid or *.midi) in either Format One or Format Two by that software.

3.2 Results

Under different parameters, I make a serial tests (see Table 15). As expected, the melodic contours of all the six experimental results (without any selection) display the effect of conservation of energy by the method in this thesis (see Figure 25 ~ Figure 32 and Table 16 ~ Table 21).

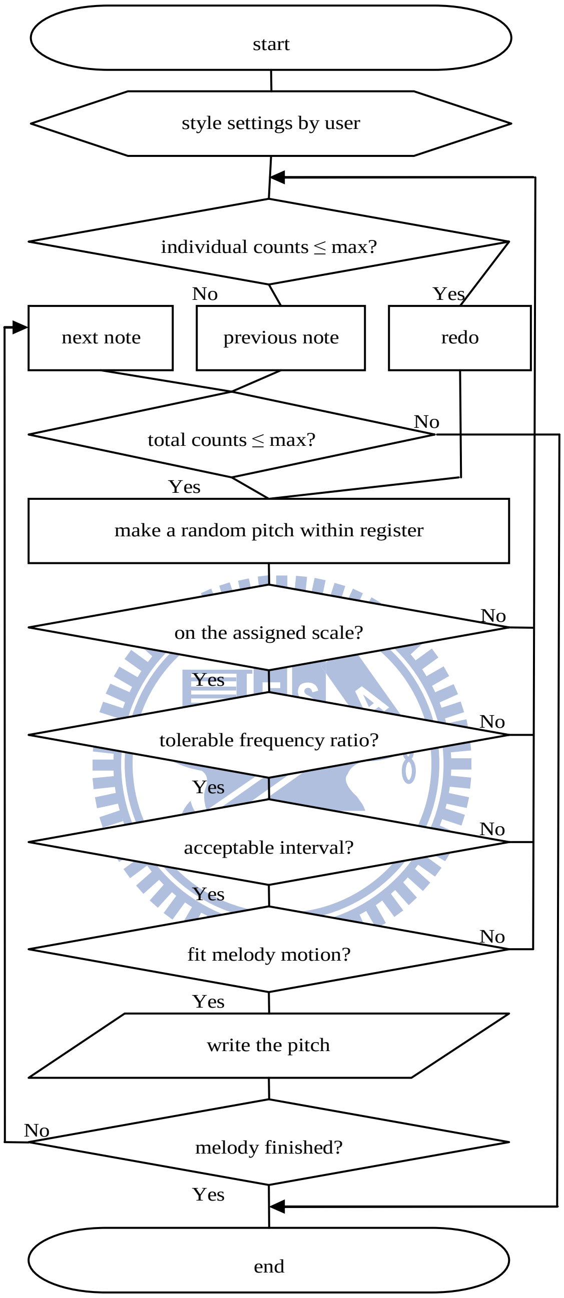


Figure 22. System structure

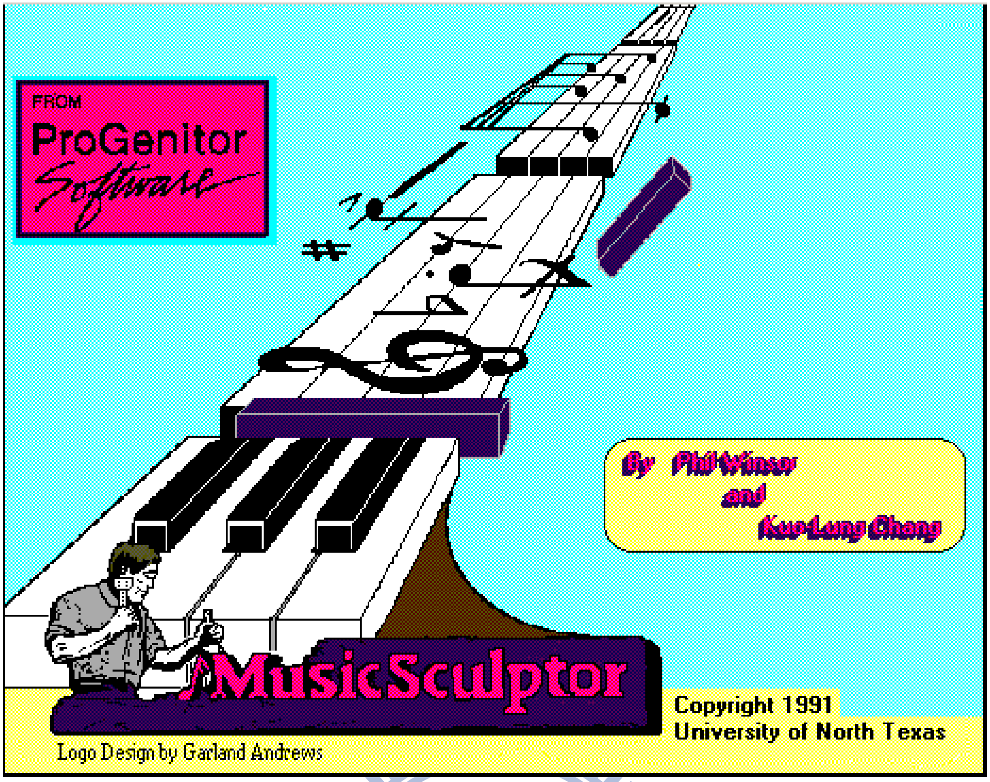
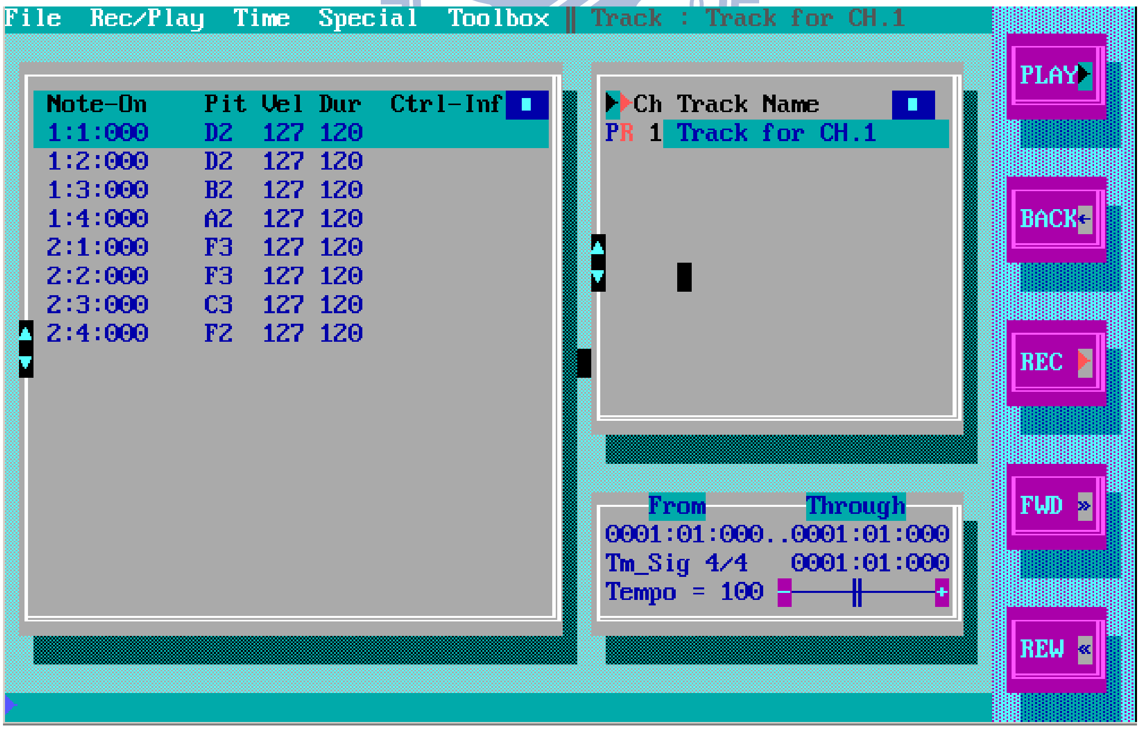


Figure 23. MusicSculptor screenshot 1



```
Note-On     Pit Vel Dur  Ctrl-Inf
1:1:000     D2  127 120
1:2:000     D2  127 120
1:3:000     B2  127 120
1:4:000     A2  127 120
2:1:000     F3  127 120
2:2:000     F3  127 120
2:3:000     C3  127 120
2:4:000     F2  127 120
```

```
Ch Track Name
PR  1 Track for CH.1
```

```
   From           Through
0001:01:000..0001:01:000
Tm_Sig 4/4    0001:01:000
Tempo  = 100
```

Figure 24. MusicSculptor screenshot 2

Table 15. Parameters of experimental results

Result	Amount	Bottom	Range	Mode	Tonic	Duration	Center	S	CSStep	CCStep	CStep	CSkip	Repetition	Count
1	12	48	24	1	0	120	60	3	4	1	5	0	0	8
2	12	48	24	1	0	120	60	3	4	1	5	1	0	8
3	12	48	24	1	0	120	60	4	4	1	5	0	0	8
4	12	48	24	1	0	120	60	4	4	1	5	1	0	8
5	10	45	24	2	9	120	57	1	4	1	5	0	0	8
6	10	45	24	2	9	120	57	1	4	1	5	1	0	8

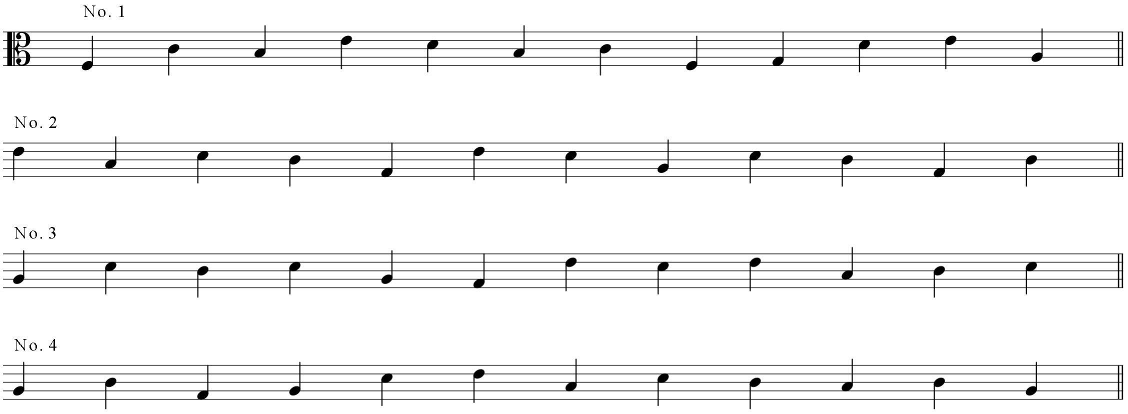



Figure 25. Experimental results 1~4

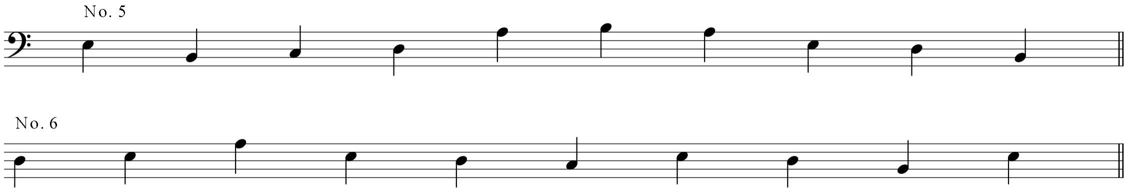


Figure 26. Experimental results 5~6

Table 16. Analysis of experimental result 1 (S=3)

Pitch	**(do4)**	**fa3**	**do4**	**si3**	**mi4**	**re4**	**si3**	**do4**	**fa3**	**sol3**	**re4**	**mi4**	**la3**
f	240	160	240	225	300	270	225	240	160	180	270	300	200
r	1.000	0.667	1.000	0.938	1.250	1.125	0.938	1.000	0.667	0.750	1.125	1.250	0.833
e	1.000	0.444	1.000	0.879	1.563	1.266	0.879	1.000	0.444	0.563	1.266	1.563	0.694
i		-0.556	0.556	-0.121	0.684	-0.297	-0.387	0.121	-0.556	0.118	0.703	0.297	-0.868
e-max		4.000	5.667	4.000	4.363	2.313	3.203	4.363	4.000	5.667	5.313	3.203	2.313
1/e-min		4.000	2.333	4.000	3.637	5.688	4.797	3.637	4.000	2.333	2.688	4.797	5.688
r-max		2.000	2.380	2.000	2.089	1.521	1.790	2.089	2.000	2.380	2.305	1.790	1.521
1/r-min		2.000	1.528	2.000	1.907	2.385	2.190	1.907	2.000	1.528	1.639	2.190	2.385
R	0.000	-0.333	0.000	-0.063	0.250	0.125	-0.063	0.000	-0.333	-0.250	0.125	0.250	-0.167
R-max		1.000	1.380	1.000	1.089	0.521	0.790	1.089	1.000	1.380	1.305	0.790	0.521
1/R-min		-1.000	-0.528	-1.000	-0.907	-1.385	-1.190	-0.907	-1.000	-0.528	-0.639	-1.190	-1.385

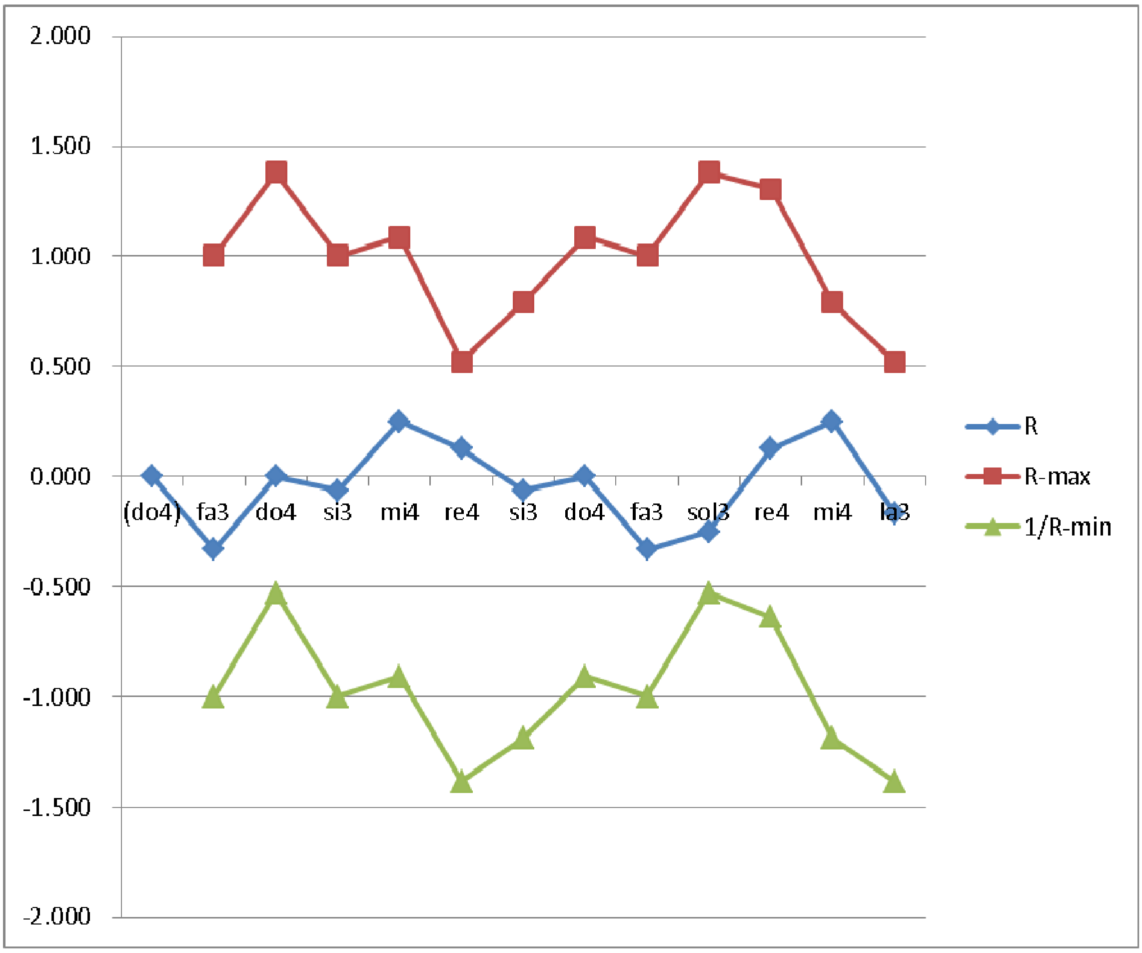


Figure 27. Chart of Table 16

Table 17. Analysis of experimental result 2 (S=3)

Pitch	**(do4)**	**mi4**	**si3**	**re4**	**do4**	**sol3**	**mi4**	**re4**	**la3**	**re4**	**do4**	**sol3**	**do4**
f	240	300	225	270	240	180	300	270	200	270	240	180	240
r	1.000	1.250	0.938	1.125	1.000	0.750	1.250	1.125	0.833	1.125	1.000	0.750	1.000
e	1.000	1.563	0.879	1.266	1.000	0.563	1.563	1.266	0.694	1.266	1.000	0.563	1.000
i		0.563	-0.684	0.387	-0.266	-0.438	1.000	-0.297	-0.571	0.571	-0.266	-0.438	0.438
e-max		4.000	2.313	4.363	3.203	4.000	5.313	2.313	3.203	4.917	3.203	4.000	5.313
1/e-min		4.000	5.688	3.637	4.797	4.000	2.688	5.688	4.797	3.083	4.797	4.000	2.688
r-max		2.000	1.521	2.089	1.790	2.000	2.305	1.521	1.790	2.217	1.790	2.000	2.305
1/r-min		2.000	2.385	1.907	2.190	2.000	1.639	2.385	2.190	1.756	2.190	2.000	1.639
R	0.000	0.250	-0.063	0.125	0.000	-0.250	0.250	0.125	-0.167	0.125	0.000	-0.250	0.000
R-max		1.000	0.521	1.089	0.790	1.000	1.305	0.521	0.790	1.217	0.790	1.000	1.305
1/R-min		-1.000	-1.385	-0.907	-1.190	-1.000	-0.639	-1.385	-1.190	-0.756	-1.190	-1.000	-0.639

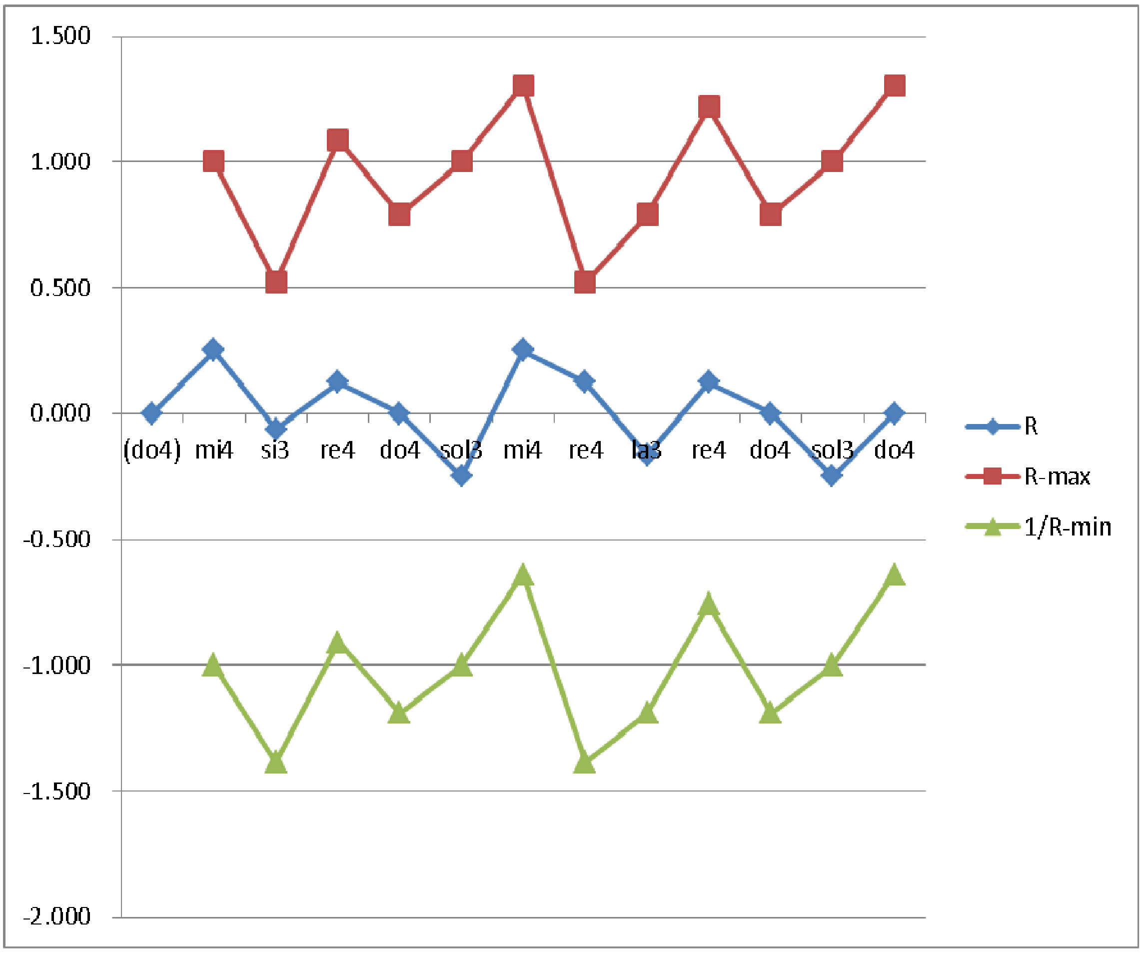


Figure 28. Chart of Table 17

Table 18. Analysis of experimental result 3 (S=4)

Pitch	**(do4)**	**la3**	**re4**	**do4**	**re4**	**la3**	**sol3**	**mi4**	**re4**	**mi4**	**si3**	**do4**	**re4**
f	240	200	270	240	270	200	180	300	270	300	225	240	270
r	1.000	0.833	1.125	1.000	1.125	0.833	0.750	1.250	1.125	1.250	0.938	1.000	1.125
e	1.000	0.694	1.266	1.000	1.266	0.694	0.563	1.563	1.266	1.563	0.879	1.000	1.266
i		-0.306	0.571	-0.266	0.266	-0.571	-0.132	1.000	-0.297	0.297	-0.684	0.121	0.266
e-max		4.000	5.222	2.938	4.000	2.938	5.222	5.750	1.750	2.938	1.750	4.484	4.000
1/e-min		4.000	2.778	5.063	4.000	5.063	2.778	2.250	6.250	5.063	6.250	3.516	4.000
r-max		2.000	2.285	1.714	2.000	1.714	2.285	2.398	1.323	1.714	1.323	2.118	2.000
1/r-min		2.000	1.667	2.250	2.000	2.250	1.667	1.500	2.500	2.250	2.500	1.875	2.000
R	0.000	-0.167	0.125	0.000	0.125	-0.167	-0.250	0.250	0.125	0.250	-0.063	0.000	0.125
R-max		1.000	1.285	0.714	1.000	0.714	1.285	1.398	0.323	0.714	0.323	1.118	1.000
1/R-min		-1.000	-0.667	-1.250	-1.000	-1.250	-0.667	-0.500	-1.500	-1.250	-1.500	-0.875	-1.000

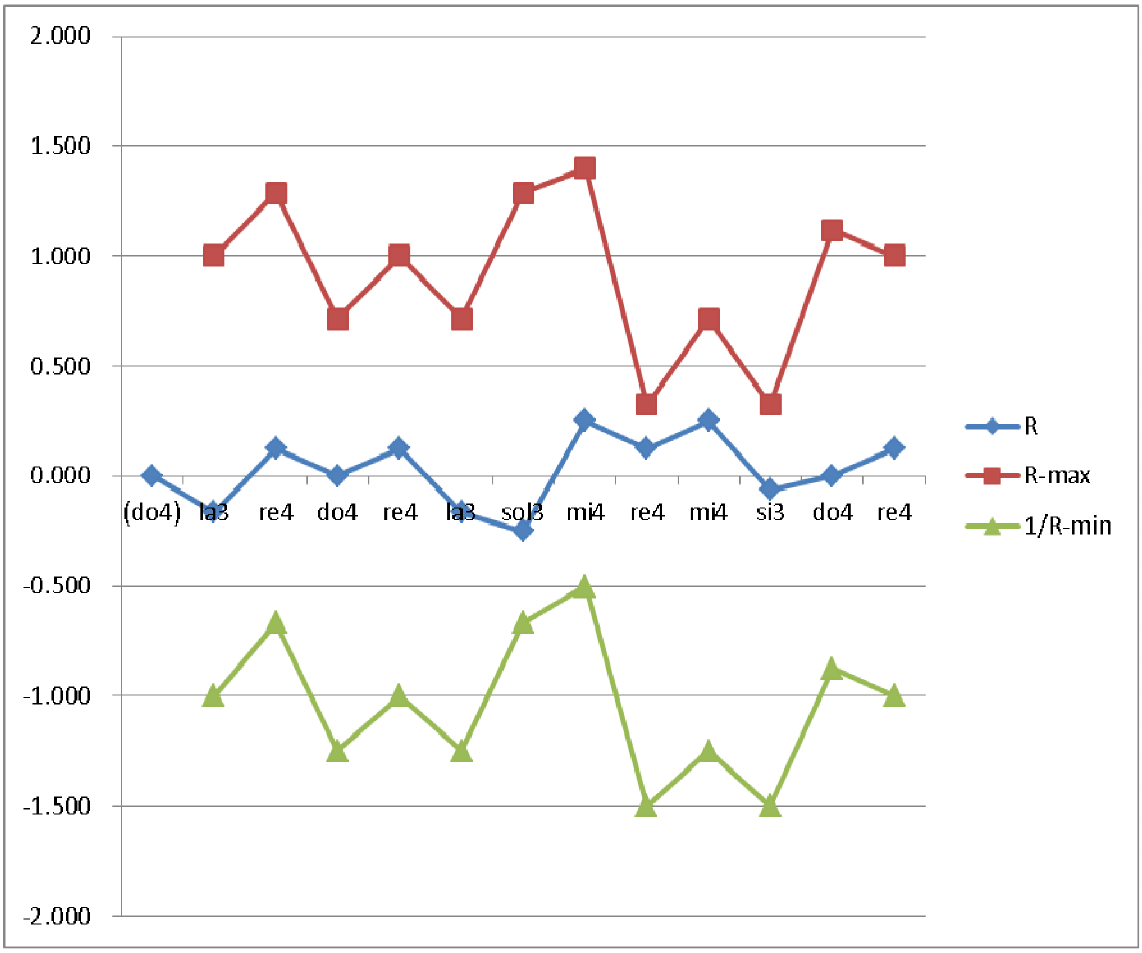


Figure 29. Chart of Table 18

Table 19. Analysis of experimental result 4 (S=4)

Pitch	**(do4)**	**la3**	**do4**	**sol3**	**la3**	**re4**	**mi4**	**si3**	**re4**	**do4**	**si3**	**do4**	**la3**
f	240	200	240	180	200	270	300	225	270	240	225	240	200
r	1.000	0.833	1.000	0.750	0.833	1.125	1.250	0.938	1.125	1.000	0.938	1.000	0.833
e	1.000	0.694	1.000	0.563	0.694	1.266	1.563	0.879	1.266	1.000	0.879	1.000	0.694
i		-0.306	0.306	-0.438	0.132	0.571	0.297	-0.684	0.387	-0.266	-0.121	0.121	-0.306
e-max		4.000	5.222	4.000	5.750	5.222	2.938	1.750	4.484	2.938	4.000	4.484	4.000
1/e-min		4.000	2.778	4.000	2.250	2.778	5.063	6.250	3.516	5.063	4.000	3.516	4.000
r-max		2.000	2.285	2.000	2.398	2.285	1.714	1.323	2.118	1.714	2.000	2.118	2.000
1/r-min		2.000	1.667	2.000	1.500	1.667	2.250	2.500	1.875	2.250	2.000	1.875	2.000
R	0.000	-0.167	0.000	-0.250	-0.167	0.125	0.250	-0.063	0.125	0.000	-0.063	0.000	-0.167
R-max		1.000	1.285	1.000	1.398	1.285	0.714	0.323	1.118	0.714	1.000	1.118	1.000
1/R-min		-1.000	-0.667	-1.000	-0.500	-0.667	-1.250	-1.500	-0.875	-1.250	-1.000	-0.875	-1.000

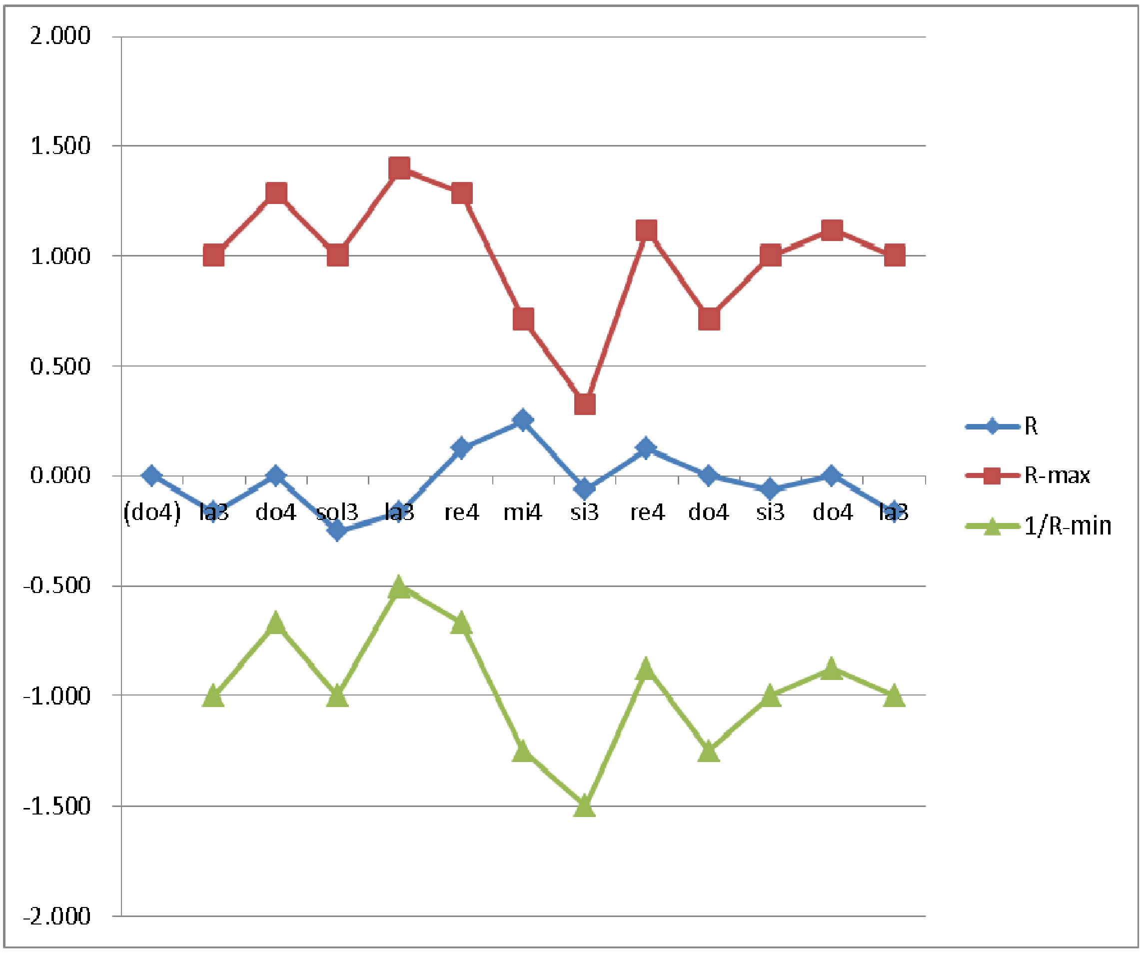


Figure 30. Chart of Table 19

Table 20. Analysis of experimental result 5 (S=1)

Pitch	**(la3)**	**mi3**	**si2**	**do3**	**re3**	**la3**	**si3**	**la3**	**mi3**	**re3**	**si2**
f	200	180	135	144	160	240	270	270	180	160	135
r	1.000	0.900	0.675	0.720	0.800	1.200	1.350	1.350	0.900	0.800	0.675
e	1.000	0.810	0.456	0.518	0.640	1.440	1.823	1.823	0.810	0.640	0.456
i		-0.190	-0.354	0.063	0.122	0.800	0.383	0.000	-1.013	-0.170	-0.184
e-max		4.000	4.190	4.544	4.482	4.360	3.560	3.178	3.178	4.190	4.360
1/e-min		4.000	3.810	3.456	3.518	3.640	4.440	4.823	4.823	3.810	3.640
r-max		2.000	2.047	2.132	2.117	2.088	1.887	1.783	1.783	2.047	2.088
1/r-min		2.000	1.952	1.859	1.876	1.908	2.107	2.196	2.196	1.952	1.908
R	0.000	-0.100	-0.325	-0.280	-0.200	0.200	0.350	0.350	-0.100	-0.200	-0.325
R-max		1.000	1.047	1.132	1.117	1.088	0.887	0.783	0.783	1.047	1.088
1/R-min		-1.000	-0.952	-0.859	-0.876	-0.908	-1.107	-1.196	-1.196	-0.952	-0.908

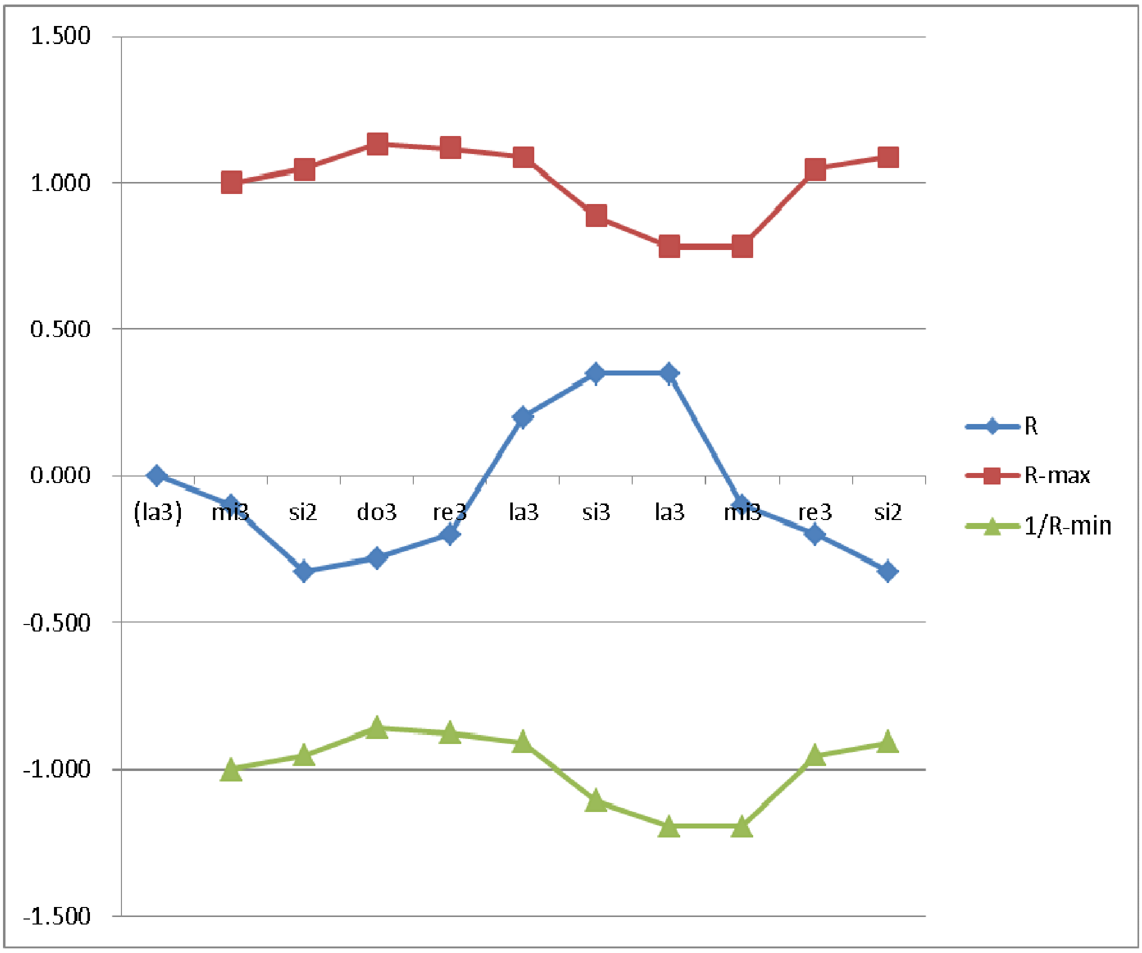


Figure 31. Chart of Table 20

Table 21. Analysis of experimental result 6 (S=1)

Pitch	**(la3)**	**re3**	**mi3**	**la3**	**mi3**	**re3**	**do3**	**mi3**	**re3**	**si2**	**mi3**
f	200	160	180	240	180	160	144	180	160	135	180
r	1.000	0.800	0.900	1.200	0.900	0.800	0.720	0.900	0.800	0.675	0.900
e	1.000	0.640	0.810	1.440	0.810	0.640	0.518	0.810	0.640	0.456	0.810
i		-0.360	0.170	0.630	-0.630	-0.170	-0.122	0.292	-0.170	-0.184	0.354
e-max		4.000	4.360	4.190	3.560	4.190	4.360	4.482	4.190	4.360	4.544
1/e-min		4.000	3.640	3.810	4.440	3.810	3.640	3.518	3.810	3.640	3.456
r-max		2.000	2.088	2.047	1.887	2.047	2.088	2.117	2.047	2.088	2.132
1/r-min		2.000	1.908	1.952	2.107	1.952	1.908	1.876	1.952	1.908	1.859
R	0.000	-0.200	-0.100	0.200	-0.100	-0.200	-0.280	-0.100	-0.200	-0.325	-0.100
R-max		1.000	1.088	1.047	0.887	1.047	1.088	1.117	1.047	1.088	1.132
1/R-min		-1.000	-0.908	-0.952	-1.107	-0.952	-0.908	-0.876	-0.952	-0.908	-0.859

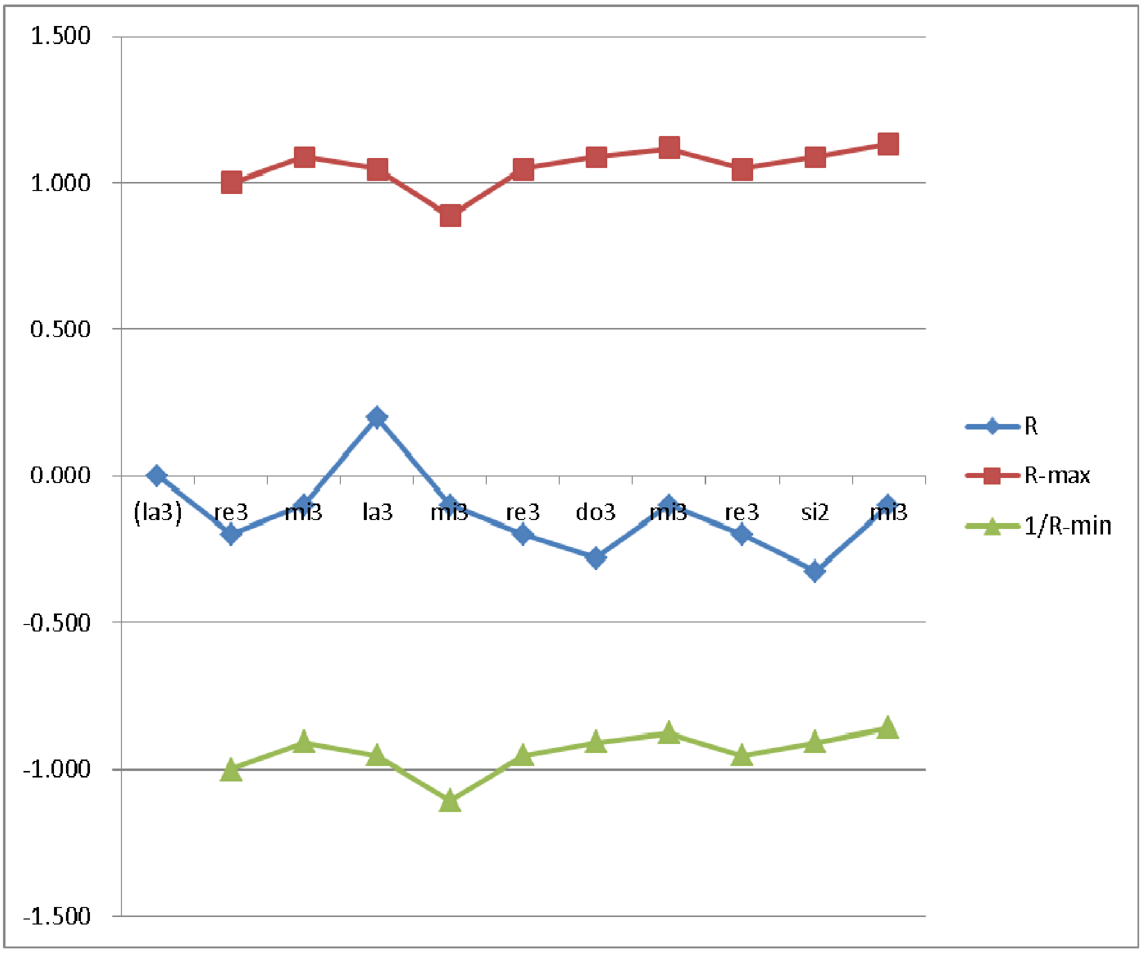


Figure 32. Chart of Table 21

4. Discussion

4.1 Interpretation

Since the pitches are randomly generated without any distribution control, any tonality or motive in the melody should be considered as a coincidence. After all, my dedication is to the melodic contour and the motion of melody pitches. Moreover, although a following unison pitch is usually an totally agreeable solution according to the model from the conservation of energy, the restriction setting is very helpful to prevent too many repetitions. There is no successive identical pitches in all the results because of the maximum restriction by the parameter settings.

A pitch sequence occasionally exists in the initial four notes of the first result. In the last five notes, the melody climbs a major seven and returns in time. In fact, it have the capability to rise more since there is still a potential space. If it continues to ascend, however, it will return at some point finally because of the effect by the model in this thesis,.

The second result shows a fair range of motion within the register parameter. It turns back dramatically after note 6 and resemble the artistic cadencial melody pattern accidentally in the last four pitches.

In the third result, note 2~4 forms an incidental auxiliary pitch formation. Moreover, the analysis reveals that the critical point is note 8, where the frequency ratio (1.125) is close to the tolerable maximum (1.323). It Goes back to the previous pitch (as note 7) forming another auxiliary pitch formation in note 7~9 and immediately skips downward to note 10 acting the vivid potential release.

Starting and ending both at “la”, the fourth result sounds like a melody on Aeolian mode rather than Ionian mode. It results from the lack of focal pitch control. Moreover, if we add a leading-tone in the ending, it will aid users to recognize the different modes.

The fifth result successfully spans a whole octave. Since the susceptibility value is merely “1”, the melody has more possibility of a huge skip. Even so, it fortuitously does not use much potential energy at any note during the whole piece. Hence we do not see a direct octave skip even in this result.

There are four identical “mi” in the ten notes of the sixth result. They mold a sporadic focal pitch. Relatively speaking, its motion and register are more conservative than previous result despite their susceptibility values are identical.

4.2 Limitation

Generally speaking, steps are insufficient in all the experimental results. They are very common in vocal melodies. It may result from the absence of an adequate distribution control. If we give steps the higher probability than skips, the problem will be less obvious.

On the other hand, I only focus on the melodic contour and the motion of melody pitch. Thereby it definitely lacks the rhythmic elements in the melody. “The connection between rhythm and melody is so close that in practice it is dangerous to try and isolate them from each other. Nine times out of ten when we speak of ‘melody’ we mean ‘rhythmic melody’; that is to say a series of notes whose mutual relationship is considered in terms of accent and duration. (Morris, 1922, 28)”

“A guaranteed procedure for the production of beautiful melodies does not exist (fortunately!). (Jeppesen, 1992, 110)” Every method of algorithmic composition is merely a part of that procedure. By only using the model in this thesis, we can guaranteed a conservation of energy in melodic contours. In order to accomplish a more specific melody, we need supplementary algorithm. For example, with the object of producing motive or pattern of pitch and rhythm, we may combine other distribution or sequence operation.

4.3 Application

Some parameters can be flexible. First, a dynamic susceptibility can create various desires for the return right after a large skip. Second, changes of the center (pitch) frequency may lead a hint of different comfortable registers. Third, if we pour (or drain) extra energy to (or from) one or both of the tolerable energy ratio variables during the piece, we will have varied fluidness of the melodic contour.

With the flexibility above, we may let part of style parameters be “fuzzy” options (Kosko, 1993 and Zimmermann, 1996) rather than exact ones. For instance, although the fifth and the sixth experimental result have the same susceptibility value, their melody motions are not alike. Nonetheless, we still can say they both have a “low” susceptibility compared with the other experimental results which have “high” susceptibilities in chapter three.

4.4 Expectation

“Creativity relies on connecting differing but viable ideas in unique and unexpected ways. (Cope, 2005, 3)” With the method in this thesis, we have prospects of connecting it with other ideas in many ways.

First of all, so far as duration and rhythmic elements are concerned, this project has nothing to report. The concept of potential energy and kinetic energy can be applied to pitch duration. Moreover, especially in counterpoint, not only horizontal melodic motions dominate, but vertical interval tensions also play another critical role. We can borrow the law of conservation of energy to examine its inner relationship from alternative perspective.

Second, the appreciation of intervallic quality and tonal harmony based on Just Intonation may be utilized as well in the domains ranging from melody analysis (e.g. Hewlett, 1998) to algorithmic composition.

Next, an automatic pitch conjecture function for playing monophonic melody through a narrow interface (e.g. Lin, 2009) could be useful at times. We can benefit from the technology when typing musical scores by computer, too.

Last but not least, with an eye to more sophisticated perspectives on exquisite music, we need more researchers to pay much attention to the frequency and energy relationship and the essence of melody composing.

4.5 Conclusion

The implementation program runs successfully; the original purpose of this thesis is basically realized accordingly. I propose a model for algorithmic composition of monophonic pitches; the melodic contour corresponds with the law of conservation of energy. In spite of the main process of energy regulation, the input and output data in this thesis are symbolic musical pitches. They can also be, however, sub-symbolic audio frequencies. We could further integrate the method with other supplementary algorithm.

I should reiterate that to reconsider the phenomena of higher levels from the point of view which we have in lower levels can assist us in composing genuinely but with imitations by adopting the principles in old styles. Above all, the domain of academic knowledge may separate again after lengthy union while it is now on the road to unite. This thesis may be a tiny part in the giant progress; nevertheless, it strongly stands for my response to the revival.

Appendix: Source Code

Visual C++ Files

stdafx.h

```
#pragma once
#include <cmath> // for pow()
#include <time.h> // for srand(time(NULL)) and rand()
#include <fstream> // for std::ofstream
```

Form1.h (excerpt)

```
#pragma once
#include "Note.h"
#include "Score.h"

System::Void button_execute_Click(System::Object^  sender, System::EventArgs^  e)
{
	short nNote = this->trackBar_nNote->Value;
	short bottom = this->trackBar_bottom->Value;
	short range= this->trackBar_range->Value;
	short mode = this->trackBar_mode->Value;
	short tonic = this->trackBar_tonic->Value;
	short duration= this->trackBar_duration->Value;
	short center= this->trackBar_center->Value;
	short susceptibility = this->trackBar_susceptibility->Value;
	short similiarStepMax = this->trackBar_similiarStepMax->Value;
	short contraryStepMax = this->trackBar_contraryStepMax->Value;
	short stepMax = this->trackBar_stepMax->Value;
	short skipMax = this->trackBar_skipMax->Value;
	short repetitionMax = this->trackBar_repetitionMax->Value;
	short exponent = this->trackBar_countMax->Value;
	double countMax = pow(2.0, exponent);
	// main function:
	short* pitches = new short[nNote]; // pitch array
```

```
    Note note(nNote, bottom, range, mode, tonic, center, susceptibility,
            similiarStepMax, contraryStepMax, stepMax, skipMax,
            repetitionMax, countMax);
    if (isComplete(nNote, pitches, countMax, note))
        finalScore(nNote, pitches, duration); // write the result score
    else
        finalScore(); // write a blank score
    delete [] pitches;
}
bool isComplete(short nNote, short* pitches, double countMax, Note note)
{
    bool is = false;
    double count = 0;
    short i = 0;
    do {
        if (i < 0) // 0. restart
            i =0;
        else if  (i == 0) // 1. first pitch
            pitches[i] = note.compose();
        else if (i == 1) // 2. second pitch
            pitches[i] = note.compose(pitches[0]);
        else // 3. following pitches
            pitches[i] = note.compose(pitches[i-2], pitches[i-1]);
        // backtrack procedure:
        if (pitches[i] < 0) // 1. rejectee: nextPitch_ == -1
            i--; // undo the previous pitch
        else // 2. pass
            i++; // do the next pitch
        // main count limit:
        count++;
        if (count > countMax)
        {
            is = false;
            break;
        }
        else
            is = true;
```

```
    }
    while (i<nNote);
    return is;
}
void finalScore(short nNote, short* pitches, short duration) // write the result score
{
    short noteon = 0;
    short pitch = 0;
    Score score;
    score.create();
    for (short i=0; i<nNote; i++){
        pitch = pitches[i];
        Score score(noteon, pitch, 127, 120, 0);
        // Noteon, Pitch, Velocity=127, Artdur=120, Channel=0
        score.write();
        noteon += duration; // whole note == 480
    }
    score.finish();
}
void finalScore() // write a blank score
{
    Score score;
    score.create();
    score.write();
    score.finish();
}
```

Standard C++ Files

Note.h

```
// Note Controller
#pragma once
#include "Motion.h"

class Note
{
```

```
public:
    Note();
    Note(short nNote, short bottom, short range, short mode,
         short tonic, short centerPitch, short susceptibility,
         short consecutiveSimiliarStepTimesMax,
         short consecutiveContraryStepMax,
         short consecutiveStepTimesMax, short consecutiveSkipTimesMax,
         short repetitionTimesMax, double countMax);
    short compose(); // first note creator
    short compose(short firstPitch); // second note creator
    short compose(short previousPitch, short presentPitch);
    // following note creator
    bool isOnModeScale(); // pitch check: mode scale
    bool isTolerableEnergyRatio(); // pitch check: energy ratio
    bool isAcceptableInterval(); // pitch check: interval
    bool isWithinRepetitionLimit(); // repetition times check
    bool isWithinConsecutiveMotionLimit(); // consecutive motion check
    void countRepetitionTimes(); // repetition times counter
    void countConsecutiveMotion(Motion motion); // consecutive motion counter
private:
    short nNote_; // note amount
    short bottom_; // bottom pitch
    short range_; // melody range
    short mode_; // mode: 1.Ionian, 2.Aeolian
    short tonic_; // tonic pitch
    short centerPitch_; // center pitch
    short susceptibility_; // susceptibility constant
    short consecutiveSimiliarStepTimesMax_; // consecutive similiar step maximum
    short consecutiveContraryStepTimesMax_; // consecutive contrary step maximum
    short consecutiveStepTimesMax_; // consecutive step maximum
    short consecutiveSkipTimesMax_; // consecutive skip maximum
    short repetitionTimesMax_; // repetition maximum
    double countMax_; // count times maximum
    short previousPitch_; // the previous pitch
    short presentPitch_; // the present pitch
    short nextPitch_; // the next pitch
    double nextEnergyRatio_; // next energy ratio
```

```
        double energyRatioMax_; // tolerable energy ratio maximum
        double energyRatioMin_; // tolerable energy ratio minmum
        double tempEnergyRatioMax_; // temporary tolerable energy ratio maximum
        double tempEnergyRatioMin_; // temporary tolerable energy ratio minmum
        short nextInterval_; // next pitch interval
        short consecutiveSimiliarStepTimes_; // consecutive similiar step times
        short consecutiveContraryTimes_; // consecutive contrary step times
        short consecutiveStepTimes_; // consecutive step times
        short consecutiveSkipTimes_; // consecutive skip times
        short repetitionTimes_; // repetition times
        short tempConsecutiveSimiliarStepTimes_;
        // temporary consecutive similiar step times
        short tempConsecutiveContraryStepTimes_;
        // temporary consecutive contrary step times
        short tempConsecutiveStepTimes_; // temporary consecutive step times
        short tempConsecutiveSkipTimes_; // temporary consecutive skip times
        short tempRepetitionTimes_; // temporary repetition times
};
```

Note.cpp

```
#include "stdafx.h"
#include "Note.h"
#include "Frequency.h"

Note::Note()
{
    srand(time(NULL));
    nNote_ = 0;
    bottom_ = 0;
    range_= 0;
    mode_ = 0;
    tonic_ = 0;
    centerPitch_= 0;
    susceptibility_ = 0;
    consecutiveSimiliarStepTimesMax_ = 0;
    consecutiveContraryStepTimesMax_ = 0;
```

```
    consecutiveStepTimesMax_ = 0;
    consecutiveSkipTimesMax_ = 0;
    repetitionTimesMax_ = 0;
    countMax_ = 0;
    previousPitch_ = 0;
    presentPitch_ = 0;
    nextPitch_ = 0;
    nextEnergyRatio_ = 0;
    energyRatioMax_ = 0;
    energyRatioMin_ = 0;
    tempEnergyRatioMax_ = 0;
    tempEnergyRatioMin_ = 0;
    nextInterval_ = 0;
    consecutiveSimiliarStepTimes_ = 0;
    consecutiveContraryTimes_ = 0;
    consecutiveStepTimes_ = 0;
    consecutiveSkipTimes_ = 0;
    repetitionTimes_ = 0;
    tempConsecutiveSimiliarStepTimes_ = 0;
    tempConsecutiveContraryStepTimes_ = 0;
    tempConsecutiveStepTimes_ = 0;
    tempConsecutiveSkipTimes_ = 0;
    tempRepetitionTimes_ = 0;
}
Note::Note(short nNote, short bottom, short range, short mode, short tonic,
           short centerPitch, short susceptibility,
           short consecutiveSimiliarStepTimesMax, short consecutiveContraryStepMax,
           short consecutiveStepTimesMax, short consecutiveSkipTimesMax,
           short repetitionTimesMax, double countMax)
 {
    srand(time(NULL));
    nNote_ = nNote;
    bottom_ = bottom;
    range_= range;
    mode_ = mode;
    tonic_ = tonic;
    centerPitch_= centerPitch;
```

```
    susceptibility_ = susceptibility;
    consecutiveSimiliarStepTimesMax_ = consecutiveSimiliarStepTimesMax;
    consecutiveContraryStepTimesMax_ = consecutiveContraryStepMax;
    consecutiveStepTimesMax_ = consecutiveStepTimesMax;
    consecutiveSkipTimesMax_ = consecutiveSkipTimesMax;
    repetitionTimesMax_ = repetitionTimesMax;
    countMax_ = countMax;
    previousPitch_ = 0;
    presentPitch_ = 0;
    nextPitch_ = 0;
    nextEnergyRatio_ = 0;
    energyRatioMax_ = 4.0; // initial Max = 4
    energyRatioMin_ = 0.25; // initial rMin = 4, therefore Min = 1 / 4
    tempEnergyRatioMax_ = 0;
    tempEnergyRatioMin_ = 0;
    nextInterval_ = 0;
    consecutiveSimiliarStepTimes_ = 0;
    consecutiveContraryTimes_ = 0;
    consecutiveStepTimes_ = 0;
    consecutiveSkipTimes_ = 0;
    repetitionTimes_ = 0;
    tempConsecutiveSimiliarStepTimes_ = 0;
    tempConsecutiveContraryStepTimes_ = 0;
    tempConsecutiveStepTimes_ = 0;
    tempConsecutiveSkipTimes_ = 0;
    tempRepetitionTimes_ = 0;
}

short Note::compose() // the first pitch
{
    presentPitch_ = centerPitch_;
    // present pitch is the center pitch; next pitch is the first pitch
    double count = 0;
    do
    {
        do nextPitch_ = rand() % range_ + bottom_;
        while (!isOnModeScale());
```

```
        Frequency frequency(mode_, tonic_, susceptibility_,
                            centerPitch_, presentPitch_, nextPitch_);
        nextEnergyRatio_ = frequency.getNextEnergyRatio();
        tempEnergyRatioMax_ = frequency.countEnergyRatioMax(energyRatioMax_);
        tempEnergyRatioMin_ = frequency.countEnergyRatioMin(energyRatioMin_);
        // backtrack procedure:
        count++;
        if (count > countMax_)
        {
            nextPitch_ = -1; // rejectee
            break;
        }
        else;
    }
    while (!isTolerableEnergyRatio());
    energyRatioMax_ = tempEnergyRatioMax_;
    energyRatioMin_ = tempEnergyRatioMin_;
    return nextPitch_;
}
short Note::compose(short presentPitch) // the second pitch
{
    presentPitch_ = presentPitch;
    double count = 0;
    do
    {
        do nextPitch_ = rand() % range_ + bottom_;
        while (!isOnModeScale());
        Frequency frequency(mode_, tonic_, susceptibility_,
                            centerPitch_, presentPitch_, nextPitch_);
        nextEnergyRatio_ = frequency.getNextEnergyRatio();
        tempEnergyRatioMax_ = frequency.countEnergyRatioMax(energyRatioMax_);
        tempEnergyRatioMin_ = frequency.countEnergyRatioMin(energyRatioMin_);
        countRepetitionTimes();
        // backtrack procedure:
        count++;
        if (count > countMax_)
        {
```

```
            nextPitch_ = -1; // rejectee
            break;
        }
        else;
    }
    while (!isTolerableEnergyRatio() || !isAcceptableInterval()
           || !isWithinRepetitionLimit());
    energyRatioMax_ = tempEnergyRatioMax_;
    energyRatioMin_ = tempEnergyRatioMin_;
    repetitionTimes_ = tempRepetitionTimes_;
    return nextPitch_;
}
short Note::compose(short previousPitch, short presentPitch) // following pitches
{
    previousPitch_ = previousPitch;
    presentPitch_ = presentPitch;
    double count = 0;
    do
    {
        do nextPitch_ = rand() % range_ + bottom_;
        while (!isOnModeScale());
        // reset variables of consecutive motion:
        tempConsecutiveSimiliarStepTimes_ = consecutiveSimiliarStepTimes_;
        tempConsecutiveContraryStepTimes_ = consecutiveContraryTimes_;
        tempConsecutiveStepTimes_ = consecutiveStepTimes_;
        tempConsecutiveSkipTimes_ = consecutiveSkipTimes_;
        tempRepetitionTimes_ = repetitionTimes_;
        // contiune:
        Frequency frequency(mode_, tonic_, susceptibility_,
                            centerPitch_, presentPitch_, nextPitch_);
        nextEnergyRatio_ = frequency.getNextEnergyRatio();
        tempEnergyRatioMax_ = frequency.countEnergyRatioMax(energyRatioMax_);
        tempEnergyRatioMin_ = frequency.countEnergyRatioMin(energyRatioMin_);
        Motion motion(previousPitch_, presentPitch_, nextPitch_);
        countRepetitionTimes();
        countConsecutiveMotion(motion);
        // backtrack procedure:
```

```
        count++;
        if (count > countMax_)
        {
            nextPitch_ = -1; // rejectee
            break;
        }
        else;
    }
    while (!isTolerableEnergyRatio() || !isAcceptableInterval()
          || !isWithinRepetitionLimit() || !isWithinConsecutiveMotionLimit());
    energyRatioMax_ = tempEnergyRatioMax_;
    energyRatioMin_ = tempEnergyRatioMin_;
    repetitionTimes_ = tempRepetitionTimes_;
    consecutiveSimiliarStepTimes_ = tempConsecutiveSimiliarStepTimes_;
    consecutiveContraryTimes_ = tempConsecutiveContraryStepTimes_;
    consecutiveStepTimes_ = tempConsecutiveStepTimes_;
    consecutiveSkipTimes_ = tempConsecutiveSkipTimes_;
    return nextPitch_;
}
bool Note::isOnModeScale()
{
    bool is = false;
    short pitchClass = (nextPitch_ + 12 - tonic_) % 12;
    switch (mode_)
    {
       case 1 : // Ionian
           if
             (
              pitchClass == 0  || pitchClass == 2  ||  pitchClass == 4  ||
              pitchClass == 5  || pitchClass == 7  ||  pitchClass == 9  ||
               pitchClass == 11
           )
              is = true;
           else
              is = false;
           break;
       case 2 : // Aeolian
```

```
            if
              (
                pitchClass == 0  || pitchClass == 2  || pitchClass == 3  ||
                pitchClass == 5  || pitchClass == 7  || pitchClass == 8  ||
                 pitchClass == 10
            )
                is = true;
            else
                is = false;
            break;
        default :
            is = false;
            break;
    }
    return is;
}
bool Note::isTolerableEnergyRatio()
{
     bool is = false;
     if (nextEnergyRatio_ > tempEnergyRatioMax_)
          is = false;
     else if (nextEnergyRatio_ < tempEnergyRatioMin_)
          is = false;
     else
          is = true;
     return is;
}
bool Note::isAcceptableInterval()
{
     bool is = false;
     short interval = nextPitch_ - presentPitch_;
     if (interval > 12) // exceed ascending octave
          is = false;
     else if (interval < -12) // exceed descending octave
          is = false;
     else if (interval == 6) // ascending tritone
          is = false;
```

```
     else if (interval == -6) // descending tritone
          is = false;
     else if (interval == 10) // ascending minor seven
          is = false;
     else if (interval == -10) // descending minor seven
          is = false;
     else if (interval == 11) // ascending major seven
          is = false;
     else if (interval == -11) // descending major seven
          is = false;
     else
          is = true;
     return is;
}
bool Note::isWithinRepetitionLimit()
{
    bool is = false;
     if (tempRepetitionTimes_ <= repetitionTimesMax_)
        is = true;
    else
        is = false; // over step, skip, or repetition times maximum
    return is;
}
bool Note::isWithinConsecutiveMotionLimit()
{
    bool is = false;
     if
     (
          tempConsecutiveSimiliarStepTimes_ <= consecutiveSimiliarStepTimesMax_ &&
          tempConsecutiveContraryStepTimes_ <= consecutiveContraryStepTimesMax_ &&
          tempConsecutiveStepTimes_ <= consecutiveStepTimesMax_ &&
          tempConsecutiveSkipTimes_ <= consecutiveSkipTimesMax_
     )
        is = true;
    else
        is = false; // over step, skip, or repetition times maximum
    return is;
```

```
}
void Note::countRepetitionTimes()
{
    nextInterval_ = nextPitch_ - presentPitch_;
    if (nextInterval_ == 0) // repetition
        tempRepetitionTimes_++;
    else // no repetition
        tempRepetitionTimes_ = 0;
}
void Note::countConsecutiveMotion(Motion motion)
{
    short identity = motion.identify();
    switch (identity)
    {
        case 1 : // consecutive similiar step
            tempConsecutiveSimiliarStepTimes_++;
            tempConsecutiveContraryStepTimes_ = 0;
            tempConsecutiveStepTimes_++;
            tempConsecutiveSkipTimes_ = 0;
            break;
        case 2 : // consecutive contrary step
            tempConsecutiveSimiliarStepTimes_ = 0;
            tempConsecutiveContraryStepTimes_++;
            tempConsecutiveStepTimes_++;
            tempConsecutiveSkipTimes_ = 0;
            break;
        case 3 : // consecutive skip
            tempConsecutiveSimiliarStepTimes_ = 0;
            tempConsecutiveContraryStepTimes_ = 0;
            tempConsecutiveStepTimes_ = 0;
            tempConsecutiveSkipTimes_++;
            break;
        default : // none of above
            tempConsecutiveSimiliarStepTimes_ = 0;
            tempConsecutiveContraryStepTimes_ = 0;
            tempConsecutiveStepTimes_ = 0;
            tempConsecutiveSkipTimes_ = 0;
```

```
            break;
    }
}
```

Frequency.h

```
// Pitch Frequency Giver
#pragma once

class Frequency
{
    public:
        Frequency();
        Frequency(short mode, short tonic, short susceptibility,
                 short centerPitch, short presentPitch, short nextPitch);
        double allocateFrequency(short octave, short pitchClass);
        double countEnergyRatioMax(double energyRatioMax); // Max
        double countEnergyRatioMin(double energyRatioMin); // Min = 1 / rMin
        double getNextEnergyRatio();
    private:
        short mode_; // mode: 1.Ionian, 2.Aeolian
        short tonic_; // tonic pitch
        short susceptibility_; // susceptibility constant
        short centerPitch_; // centerPitch
        short presentPitch_; // present pitch
        short nextPitch_; // next pitch
        short centerOctave_; // center pitch register location
        short presentOctave_ ; // present pitch register location
        short nextOctave_ ; // next pitch register location
        short centerPitchClass_; // center pitch class
        short presentPitchClass_; // present pitch class
        short nextPitchClass_; // next pitch class
        double centerFrequency_; // center pitch frequency
        double presentFrequency_; // present pitch frequency
        double nextFrequency_; // next pitch frequency
        double presentFrequencyRatio_;
        // present frequency ratio: frequency (n-1) / frequency (0)
```

```
        double nextFrequencyRatio_;
        // next frequency ratio: frequency (n) / frequency (0)
        double presentEnergyRatio_;
        // present energy ratio: (frequency (n-1) / frequency (0))^2
        double nextEnergyRatio_;
        // next energy ratio: (frequency (n) / frequency (0))^2
        double energyRatioInterval_; // energy ratio interval
};
```

Frequency.cpp

```
#include "stdafx.h"
#include "Frequency.h"

Frequency::Frequency()
{
    mode_ = 0;
    tonic_ = 0;
    susceptibility_ = 0;
    centerPitch_ = 0;
    presentPitch_ = 0;
    nextPitch_ = 0;
    centerPitchClass_ = 0;
    presentPitchClass_ = 0;
    nextPitchClass_ = 0;
    centerOctave_ = 0;
    presentOctave_ = 0;
    nextOctave_ = 0 ;
    centerFrequency_ = 0;
    presentFrequency_ = 0;
    nextFrequency_ = 0;
    presentFrequencyRatio_ = 0;
    nextFrequencyRatio_ = 0;
    presentEnergyRatio_ = 0;
    nextEnergyRatio_ = 0;
    energyRatioInterval_ = 0;
}
```

```
Frequency::Frequency(short mode, short tonic, short susceptibility,
                     short centerPitch, short presentPitch, short nextPitch)
 {
     mode_ = mode;
     tonic_ = tonic;
     susceptibility_ = susceptibility;
     centerPitch_ = centerPitch;
     presentPitch_ = presentPitch;
     nextPitch_ = nextPitch;
     centerPitchClass_ = (centerPitch_ + 12 - tonic_) % 12;
     presentPitchClass_ = (presentPitch_ + 12 - tonic_) % 12;
     nextPitchClass_ = (nextPitch_ + 12 - tonic_) % 12;
     centerOctave_ = (centerPitch_ + 12) / 12 - 1;
     presentOctave_ = (presentPitch_ + 12) / 12 - 1;
     nextOctave_ = (nextPitch_ + 12) / 12 - 1;
     centerFrequency_ = allocateFrequency(centerOctave_, centerPitchClass_);
     presentFrequency_ = allocateFrequency(presentOctave_, presentPitchClass_);
     nextFrequency_ = allocateFrequency(nextOctave_, nextPitchClass_);
     presentFrequencyRatio_ = presentFrequency_ / centerFrequency_;
     nextFrequencyRatio_ = nextFrequency_ / centerFrequency_;
     presentEnergyRatio_ = presentFrequencyRatio_ * presentFrequencyRatio_;
     nextEnergyRatio_ = nextFrequencyRatio_ * nextFrequencyRatio_;
     energyRatioInterval_ = nextEnergyRatio_ - presentEnergyRatio_;
}
double Frequency::allocateFrequency(short octave, short pitchClass)
{
     double frequency = 0;
     switch (mode_)
     {
          case 1 : // Ionian
               switch (pitchClass)
               {
                    case  0 : frequency = 24 * pow(2.0, octave); break;
                    case  2 : frequency = 27 * pow(2.0, octave); break;
                    case  4 : frequency = 30 * pow(2.0, octave); break;
                    case  5 : frequency = 32 * pow(2.0, octave); break;
                    case  7 : frequency = 36 * pow(2.0, octave); break;
```

```
                case  9 : frequency = 40 * pow(2.0, octave); break;
                case 11 : frequency = 45 * pow(2.0, octave); break;
                default : frequency = 1; break;
            }
            break;
        case 2 : // Aeolian
            switch (pitchClass)
            {
                case  0 : frequency = 120 * pow(2.0, octave); break;
                case  2 : frequency = 135 * pow(2.0, octave); break;
                case  3 : frequency = 144 * pow(2.0, octave); break;
                case  5 : frequency = 160 * pow(2.0, octave); break;
                case  7 : frequency = 180 * pow(2.0, octave); break;
                case  8 : frequency = 192 * pow(2.0, octave); break;
                case 10 : frequency = 216 * pow(2.0, octave); break;
                default : frequency = 2; break;
            }
            break;
        default : frequency = 3; break;
    }
    return frequency;
}
double Frequency::countEnergyRatioMax(double energyRatioMax)
{
    energyRatioMax = energyRatioMax - (energyRatioInterval_ * susceptibility_);
    // next Max
    if (energyRatioMax < 1)
        energyRatioMax = 1; // Max always >=1
    return energyRatioMax;
}
double Frequency::countEnergyRatioMin(double energyRatioMin)
{
    double reciprocalEnergyRatioMin = 1 / energyRatioMin; // present rMin
    reciprocalEnergyRatioMin = reciprocalEnergyRatioMin + (energyRatioInterval_
                               * susceptibility_); // next rMin
    if (reciprocalEnergyRatioMin < 1)
        reciprocalEnergyRatioMin = 1; // rMin always >=1
```

```
    energyRatioMin = 1 / reciprocalEnergyRatioMin; // next Min
    return energyRatioMin;
}
double Frequency::getNextEnergyRatio()
{
    return nextEnergyRatio_;
}
```

Motion.h

```
// Melody Motion Identifier
#pragma once

class Motion
{
    public:
        Motion();
        Motion(short previousPitch, short presentPitch, short nextPitch);
        short identify(); // motion types identifier
    private:
        short previousPitch_;    // 0 ~ 127
        short presentPitch_;   // 0 ~ 127
        short nextPitch_;    // 0 ~ 127
        short previousInterval_; // -127 ~ +127
        short nextInterval_; // -127 ~ +127
        bool areSimilarDirections_; // T: similiar, F: contrary
        short previousMotion_;   // 0: repetition, 1: step, 2: skip
        short nextMotion_;   // 0: repetition, 1: step, 2: skip
};
```

Motion.cpp

```
#include "stdafx.h"
#include "Motion.h"

Motion::Motion()
{
    previousPitch_ = 0;
```

```
     presentPitch_ = 0;
     nextPitch_ = 0;
     previousInterval_ = 0;
     nextInterval_ = 0;
     areSimilarDirections_ = false;
     previousMotion_ = 0;
     nextMotion_ = 0;
}
Motion::Motion(short previousPitch, short presentPitch, short nextPitch)
{
    previousPitch_ = previousPitch;
    presentPitch_ = presentPitch;
    nextPitch_ = nextPitch;
    previousInterval_ = presentPitch_ - previousPitch_;
    nextInterval_ = nextPitch_ - presentPitch_;
     // directions:
    if (previousInterval_ * nextInterval_ >= 0) // not "> 0"
        areSimilarDirections_ = true; // similiar
    else
        areSimilarDirections_ = false; // contrary
     // previous motion:
    if (previousInterval_ == 0)
        previousMotion_ = 1; // repetition
    else if (previousInterval_ <= 2 && previousInterval_ >= -2)
          previousMotion_ = 2; // step
    else
        previousMotion_ = 3; // skip
     // next motion:
    if (nextInterval_ == 0)
        nextMotion_ = 1; // repetition
    else if (nextInterval_ <= 2 && nextInterval_ >= -2)
        nextMotion_ = 2; // step
    else
        nextMotion_ = 3; // skip
}
short Motion::identify()
{
```

```
    short identity = 0;
    if (previousMotion_ == 2 && nextMotion_ == 2)
    {
        switch (areSimilarDirections_)
        {
            case true : identity = 1; break;// consecutive similiar step
            case false : identity = 2; break;// consecutive contrary step
        }
    }
    else if (previousMotion_ == 3 && nextMotion_ == 3)
        identity = 3; // consecutive skip
    else
        identity = 4; // none of above
    return identity;
}
```

Score.h

```
// Score Writer for the File Format of MusicSculptor by Phil Winsor
#pragma once

class Score
{
    public:
        Score();
        Score(short noteon, short pitch, short velocity, short artdur, short channel);
        void create(); // output file initiator
        void write(); // note writer
        void finish(); // output file closer
    private:
        short noteon_;
        short pitch_;
        short velocity_;
        short artdur_;
        short channel_;
};
```

Score.cpp

```
#include "stdafx.h"
#include "Score.h"

Score::Score()
{
    noteon_= 0;
    pitch_ = 0;
    velocity_ = 0;
    artdur_ = 0;
    channel_ = 0;
}
Score::Score(short noteon, short pitch, short velocity, short artdur, short channel)
{
    noteon_= noteon;
    pitch_ = pitch;
    velocity_ = velocity;
    artdur_ = artdur;
    channel_ = channel;
}
void Score::create()
{
    // create an output file in .sc format:
    std::ofstream tfile("Output.sc");
    // write the head of the output file:
    tfile<<"File-ID : 2"<<std::endl;
    tfile<<"Format : 1"<<std::endl;
}
void Score::write()
{
    // append the note to the output file
    std::ofstream tfile("Output.sc", std::ios::app);
    // write the note
    tfile<<noteon_<<' '<<pitch_<<' '<<velocity_<<' '<<artdur_<<' '<<channel_<<
        std::endl;
}
```

```
void Score::finish()
{
    // write the end flag of the output file:
     std::ofstream tfile("Output.sc", std::ios::app);
     tfile<<"0 -1 0 0 0"<<std::endl;
}
```

References

Adiloglu, Kamil, and Ferda N. Alpaslan. “A Machine Learning Approach to Two-Voice Counterpoint Composition.” *Knowledge-based Systems* 20, no. 3 (2007): 300-09.

Aguilera, Gabriel, José Luis Galán, Rafael Madrid, Antonio Manuel Martínez, Yolanda Padilla, and Pedro Rodríguez. “Automated Generation of Contrapuntal Musical Compositions Using Probabilistic Logic in Derive.” *Mathematics and Computers in Simulation* 80, no. 6 (2010): 1200-11.

Aldwell, Edward, and Carl Schachter. *Harmony and Voice Leading*. 3rd ed. New York: Schirmer, 2003.

Baggi, Denis, ed. *Readings in Computer-Generated Music*. Los Alamitos: IEEE Computer Society Press, 1992.

Balaban, Mira, Kemal Ebcioglu, and Otto Laske, eds. *Understanding Music with Ai: Perspectives on Music Cognition*. Cambridge, MA: The MIT Press, 1992.

Bobbitt, Richard. “The Physical Basis of Intervallic Quality and Its Application to the Problem of Dissonance.” *Journal of Music Theory* 3, no. 2 (1959): 173-207.

Brower, Candace. “Paradoxes of Pitch Space.” *Music Analysis* 27, no. 1 (2008): 51-106.

Bruner, Cheryl L. “The Perception of Contemporary Pitch Structures.” *Music Perception* 2, no. 1 (1984): 25-39.

Callender, Clifton, Ian Quinn, and Dmitri Tymoczko. “Generalized Voice-Leading Spaces.” *Science* 320, no. 5874 (2008): 346-48.

Campbell, Murray, and Clive Greated. *The Musician's Guide to Acoustics*. London: J. M. Dent & Sons Ltd, 1987.

Chew, Elaine, Adrian Childs, and Ching-Hua Chuan. “Mathematics and Computation in Music.” Paper presented at the Second international conference, MCM, New Haven, CT, USA, 2009.

Collins, Nicholas, and Nick Collins. *Introduction to Computer Music*. John Wiley & Sons, 2010.

Cooper, Grosvenor W., and Leonard B. Meyer. *The Rhythmic Structure of Music*. Chicago: University of Chicago, 1960.

Cope, David. *Computer Models of Musical Creativity*. Cambridge, MA: MIT Press, 2005.

———. *Hidden Structure: Music Analysis Using Computers*. The Computer Music and Digital Audio Series. edited by James Zychowicz Middleton: A-R Editions, 2008.

———. *Virtual Music: Computer Synthesis of Musical Style*. Cambridge, MA: The MIT Press, 2001.

Dallin, Leon. *Techniques of Twentieth Century Composition: A Guide to the Materials of Modern Music*. Music Series. 3rd ed. Dubuque: W. C. Brown Co., 1974.

Duffin, Ross W. *How Equal Temperament Ruined Harmony (and Why You Should Care)*. New York: W. W. Norton, 2007.

Farbood, Mary, and Bernd Schoner. “Analysis and Synthesis of Palestrina-Style Counterpoint Using Markov Chains.” In *International Computer Music Conference*. Habana, Cuba, 2001.

Fauvel, John, Raymond Flood, and Robin Wilson, eds. *Music and Mathematics: From Pythagoras to Fractals*. Oxford: Oxford University Press, 2006.

Foerster, Heinz Von, and James W. Beauchamp, eds. *Music by Computers*. New York: John Wiley & Sons, 1969.

Gedik, Ali C., and Barıs Bozkurt. “Pitch-Frequency Histogram-Based Music Information Retrieval for Turkish Music.” *Signal Processing* 90, no. 4 (2010): 1049-63.

Halliday, David, Robert Resnick, and Jearl Walker. *Fundamentals of Physics Extended*. 8th ed. Hoboken: John Wiley & Sons, 2008.

Harkleroad, Leon. *The Math Behind the Music*. Cambridge: Cambridge University Press, 2006.

Hasty, Christopher F. *Meter as Rhythm*. Oxford: Oxford University Press, 1997.

Helmholtz, Hermann L. F. *On the Sensations of Tone as a Physiological Basis for the Theory of Music*. New York: Dover Publications, 1954.

Hewlett, Walter B., and Eleanor Selfridge-Field, eds. *Melodic Similarity: Concepts, Procedures, and Applications*, Computing in Musicology, vol. 11. Cambridge, MA: The MIT Press, 1998.

Hiller, Lejaren A., and Leonard M. Isaacson. *Experimental Music: Composition with an Electronic Computer*. New York: McGraw-Hill, 1959.

Hsu, Kenneth J., and Andreas J. Hsu. “Fractal Geometry of Music.” *Proceedings of the National Academy of Sciences of the United States of America* 87, no. 3 (1990): 938-41.

———. “Self-Similarity of the ‘1/F Noise’ Called Music.” *Proceedings of the National Academy of Sciences of the United States of America* 88, no. 8 (1991): 3507-09.

Jeppesen, Kund. *Counterpoint: The Polyphonic Vocal Style of the Sixteenth Century*. Translated by Glen Haydon. New York: Dover Publications, 1992.

Jonhson, Timothy A. *Foundations of Diatonic Theory: A Mathematically Based Approach to Music Fundamentals*. Lanham: Scarecrow Press, 2008.

Kennan, Kent Wheeler. *Counterpoint: Based on Eighteenth-Century Practice*. 3rd ed. Englewood Cliffs: Prentice-Hall, 1987.

Kosko, Bart. *Fuzzy Thinking: The New Science of Fuzzy Logic*. New York: Hyperion, 1993.

Kostka, Stefan, and Dorothy Payne. *Tonal Harmony: With an Introduction to Twentieth-Century Music*. 4th ed. New York: McGraw-Hill, 2000.

Krumhansl, Carol L. *Cognitive Foundations of Musical Pitch*. Oxford Psychology Series. edited by Donald E. Broadbent, James L. McGaugh, Stephen Kosslyn, Endel Tulving, Nicholas J. Mackintosh and Lawrence Weisskrantz Oxford: Oxford University Press, 1990.

Lerdahl, Fred. *Tonal Pitch Space*. Oxford: Oxford University Press, 2005.

Lerdahl, Fred, and Ray Jackendoff. *A Generative Theory of Tonal Music*. Cambridge, MA: The MIT Press, 1983.

Lewin, David. *Generalized Musical Intervals and Transformations*. Oxford: Oxford University Press, 1987.

Lin, Hsin-Ming, and Chih-Fang Huang. “An Algorithmic Composition Method of Monophonic Pitches Based on the Frequency Relationship of Melody Motion.” In *International Computer Music Conference*, 518-21. Suffolk, NY, USA, 2010.

———. “A Better Layout for Playing Melody by Cellphone Keypad.” In *The 4th Intelligent Living Technology Conference*, 1555-58. Taichung, Taiwan, 2009.

Lloyd, Llewelyn Southworth, and Hugh Boyle. *Intervals, Scales and Temperaments*. New York: St.

Martin's Press, 1979.

Loy, Gareth. *Musimathics: The Mathematical Foundations of Music*. Cambridge, MA: MIT Press, 2006.

Ma, James. “Theory and Application of the Dynamic Just Intonation.” *The Periodical of Taiwanese Computer Music Association* (2002): 23-33.

Mann, Alfred. *The Study of Counterpoint: From Johann Joseph Fux's Gradus Ad Parnassum*. Revised ed. New York: W. W. Norton & Company, 1965.

Mathews, Max V., and John R. Pierce, eds. *Current Directions in Computer Music Research*, System Development Foundation Benchmark. Cambridge, MA: The MIT Press, 1991.

Mazzola, Guerino. *The Topos of Music: Geometric Logic of Concepts, Theory, and Performance* Basel: Birkhäuser, 2002.

Miranda, Eduardo Reck. *Composing Music with Computers*. Music Technology Series. edited by Francis Rumsey Burlington: Focal Press, 2001.

Morris, Robert D. *Composition with Pitch-Classes: A Theory of Compositional Design*. New Haven: Yale University Press, 1987.

———. “New Directions in the Theory and Analysis of Musical Contour.” *Music Theory Spectrum* 15, no. 2 (1993): 205-28.

———. “Voice-Leading Spaces.” *Music Theory Spectrum* 20, no. 2 (1998): 175-202.

Morris, Reginald Owen. *Contrapuntal Technique in the Sixteenth Century*. Oxford: Oxford University Press, 1922.

Nierhaus, Gerhard. *Algorithmic Composition: Paradigms of Automated Music Generation*. Vienna: SpringerWienNewYork, 2009.

Parncutt, Richard. *Harmony: A Psychoacoustical Approach*. Springer Series in Information Sciences. edited by Manfred R. Schroeder New York: Springer-Verlag, 1989.

Roads, Curtis, and John Strawn, eds. *Foundations of Computer Music*. Cambridge, MA: MIT Press, 1985.

Rossing, Thomas D., F. Richard Moore, and Paul A. Wheeler. *The Science of Sound*. 3rd ed. San

Francisco: Addison Wesley, 2002.

Rowe, Robert. *Interactive Music Systems: Machine Listening and Composing*. Cambridge, MA: The MIT Press, 1993.

———. *Machine Musicianship*. Cambridge, MA: The MIT Press, 2001.

———. “Split Levels: Symbolic to Sub-Symbolic Interactive Music Systems.” *Contemporary Music Review* 28, no. 1 (2009): 31-42.

Schell, Daniel. “Optimality in Musical Melodies and Harmonic Progressions: The Travelling Musician.” *European Journal of Operational Research* 140 (2002): 354-72.

Schmuckler, Mark. A. “Melodic Contour Similarity Using Folk Melodies.” *Music Perception* 28, no. 2 (2010): 169-93.

Schoenberg, Arnold. *8 Lieder, Op. 6*. Berlin: Verlag Dreililien (Richard Birnbach), 1908.

———. *Fundamentals of Musical Composition*. Translated by Gerald Strang. London: Faber & Faber, 1967.

Schottstaedt, Bill. “Automatic Species Counterpoint.” Stanford University, 1984.

Schwanauer, Stephan M., and David A. Levitt, eds. *Machine Models Music*. Cambridge, MA: The MIT Press, 1993.

Serra, Xavier. “Towards a Roadmap for the Research in Music Technology.” In *International Computer Music Conference*. Barcelona, Spain, 2005.

Straus, Joseph N. *Introduction to Post-Tonal Theory*. Englewood Cliffs: Prentice-Hall, 1990.

Taube, Heinrich K. *Notes from the Metalevel: Introduction to Algorithmic Music Composition*. Studies on New Music Research. edited by Marc Leman London: Taylor & Francis, 2004.

Temperley, David. “An Algorithm for Harmonic Analysis.” *Music Perception* 15, no. 1 (1997): 31-68.

———. *The Cognition of Basic Musical Structures*. Cambridge, MA: The MIT Press, 2001.

———. “The Line of Fifths.” *Music Analysis* 19, no. 3 (2000): 289-319.

———. *Music and Probability*. Cambridge, MA: The MIT Press, 2007.

Thompson, William Forde. *Music, Thought, and Feeling: Understanding the Psychology of Music*. Oxford: Oxford University Press, 2008.

Thornton, Stephen T., and Jerry B. Marion. *Classical Dynamics of Particles and Systems*. 5th ed. Belmont: Brooks Cole, 2004.

Toussaint, Godfried. “Computational Geometric Aspects of Rhythm, Melody, and Voice-Leading.” *Computational Geometry* 43 (2010): 2-22.

Tymoczko, Dmitri. “The Geometry of Musical Chords.” *Science* 313, no. 5783 (2006): 72-74.

———. “Scale Theory, Serial Theory and Voice Leading.” *Music Analysis* 27, no. 1 (2008): 1-49.

Winsor, Phil. *Automated Music Composition*. Denton: University of North Texas Press, 1989.

Winsor, Phil, and Gene De Lisa. *Computer Music in C*. Denton: University of North Texas Press, 1991.

Wu, Dian-Foon. “An Expert System for Composing 16th Century's Palestrina Musical Style.” Master's thesis, National Chiao Tung University, 1988.

Xenakis, Iannis. *Formalized Music: Thought and Mathematics in Composition*. Harmonologia Series. edited by Sharon Kanach. Revised ed. Hillsdale: Pendragon Press, 1992.

Zimmermann, Hans-Jürgen. *Fuzzy Set Theory and Its Applications*. 3rd ed. Norwell: Kluwer Academic Publisher, 1996.

Zuckerkandl, Victor. *The Sense of Music*. Corrected Edition, with index, 1971 ed. Princeton: Princeton University Press, 1971.