

Experimental Music

COMPOSITION WITH
AN ELECTRONIC COMPUTER

Lejaren A. Hiller, Jr.

ASSISTANT PROFESSOR OF MUSIC
SCHOOL OF MUSIC, UNIVERSITY OF ILLINOIS

Leonard M. Isaacson

MATHEMATICIAN
STANDARD OIL COMPANY OF CALIFORNIA
FORMERLY RESEARCH ASSOCIATE
UNIVERSITY OF ILLINOIS

MCGRAW-HILL BOOK COMPANY, INC.
1959 NEW YORK TORONTO LONDON

C H A P T E R F I V E

Experimental Details

Outline of Experiments

The central body of work, exclusive of the brief preliminary examination of the statistics of the note-selection process discussed in the last chapter, can be grouped into four principal "Experiments." In Experiment One, the two major objectives were (1) to develop a technique for the composition of a simple but recognizable type of melody and (2) to achieve simple polyphonic writing. This second objective, in particular, seemed an essential prerequisite to further experimentation of the type contemplated. Strict counterpoint was selected as the musical medium for Experiment One, which then consisted of three computer programs designed to produce successively monody, two-part writing, and, lastly, four-part writing. However, only a limited number of counterpoint rules were employed, and note selection was confined to the white notes of the C-major scale. This was done to simplify the coding in this initial experiment. Therefore, in the first part of Experiment One, a program was written to permit the writing of simple *cantus firmus* melodies varying from three to twelve notes in length. Then, two-part writing was studied in order to investigate the coding of simple contrapuntal relationships. A number of new problems arose—vertical or chordal relationships, contrary and parallel motion, the

rejection of dissonances, and the formation of a simple cadence in C at the end of each *cantus firmus* setting. Lastly, in the third part of this experiment, we were concerned primarily with three-way and four-way note interactions and, in general, with the problem of organizing a much more complex texture and sequence of operations.

At this juncture, it became apparent that all the important remaining rules of counterpoint should be added to the computer code in order to demonstrate more clearly that a recognizable type of music could be produced by computational techniques. Therefore, in what we shall call Experiment Two, a completely rewritten code was prepared to permit the generation of essentially correct four-part first-species counterpoint. This code was so designed that the rules could be added or removed from the set of instructions by simple alterations in the code. Musical output was generated which included examples of settings from purely random white-note writing, obeying no rules whatsoever, to the most restricted settings, in which the full set of rules were applied.

One possible way to continue from this point would have been to proceed to second-species counterpoint—two-against-one writing, rhythmically—then to third species, to fourth species, and eventually to florid counterpoint. In so doing, we would duplicate the experience of a music-theory student, and new and important musical problems such as elementary rhythms, weak and strong beats, and the resolution of dissonances would be introduced in a systematic way. However, it was apparent that this would involve a long process of code writing which, when completed, would still only result in an imitation of one historical musical style. Moreover, it was evident that other types of codes might provide more effective computer methods for various technical problems of music writing and, in particular, for writing music of greater contemporary stylistic interest than strict counterpoint. Therefore, in the first part of Experiment Three, a method was devised for generating independently for each voice rhythmic patterns, dynamic markings, and an index of playing instructions for stringed instruments, such as *legato*, *staccato*, *pizzicato*, and *sul ponte cello*. This last is an elementary prototype for the more general problem of orchestration. Secondly, the problem of generating the tones to be played was reexamined. Our system was expanded to include the complete chromatic scale rather than just white notes. Initially we generated purely random chromatic music in order to have examples of the most chaotic starting tonal materials, that is, materials of the highest entropy content within the framework of the chromatic scale. Then, in the third part of

this third experiment, we modified the rhythm, dynamics, and playing-instructions code so that each of these might pertain to more than one voice at a time, and thus no longer leave the coincidence of rhythm, dynamics, and color effects entirely to chance. Subsequently, this code was combined with the random-note code to provide rhythmic randomly pitched music.

The next step was to restrict the selection of notes somewhat and to give some coherence to the melodic lines. To this end several simple rules were used: (1) a rather special type of octave-range rule; (2) a rule involving the stepwise resolution of tritones to consonant intervals; and (3) a "skip-stepwise" type of melodic rule. This music was generated and then combined with the previously described code for rhythm, dynamics, and playing instructions to produce a texture freely chromatic and dissonant in style, and yet obviously under greater control than purely random output. Since all of these musical examples turned out, as expected, to be highly reminiscent, in terms of over-all sound, of certain styles of contemporary music writing, we completed this Experiment Three by developing a simple technique for generating twelve-tone rows and certain similar restrictive structural devices of current interest.

Experiments of this type, however, soon lead to the asking of whether there are formal principles of organization more general than the various rules of composition involving harmony, counterpoint, and similar techniques. This question formed the basis of studies which have been grouped together as Experiment Four. Since the technical procedure used to produce music by means of the computer was the application of the principles of Markoff processes, it was decided to generate what can be called *Markoff chain music*. A series of studies was undertaken to produce a number of examples of such music. To this end we defined two sets of numbers for assigning values to both absolute and conditional transition probabilities for successive note selection. One set of values was based on the overtone series and permitted the assignment of probabilities for melodic intervals related to their order of occurrence in the harmonic series and, hence, to their relative degrees of consonance. The second set of numbers was used to extend the idea of a leading-tone function. These values were used to control the relative proportions of larger and smaller melodic intervals. Transition probabilities derived from these two sets of integers were combined in various ways to produce melodic output in which the proportion and character of skips and stepwise motions, the proportion of consonant to dissonant intervals, and the resolution of dissonant to consonant tex-

tures, or vice versa, were controlled by rather simple means. This last experiment was completed by applying the concept of Markoff chain transition probabilities to the problem of writing an extended cadence, specifically a generalized I-IV-V-I (tonic-subdominant-dominant-tonic) cadence in C.

An outline of the experiments carried out to produce the *Illiadic Suite* is given in Table 4. This is also a chronological record of work carried out from September, 1955, through November, 1956.

Experiment One

The logic used to write a computer program differs from the logic one might ordinarily use to explain compositional problems in strictly musical terms. In planning a computer program, the first step is to design a block diagram which outlines the required logical processes. This is followed by the actual writing of the program itself. To illustrate this process, it is convenient to consider the planning of the program for four-part first-species counterpoint, the musical form subject to investigation in the last part of Experiment One and in Experiment Two. This arrangement also permits us to discuss the details of Experiment One rather sketchily since all the significant aspects of coding developed for Experiment One were later refined and incorporated into the more complex set of instructions developed for Experiment Two. In both experiments the same basic musical technique, namely, strict counterpoint, was employed. The pertinent coding details discussed under Experiment Two may, therefore, be considered to apply to Experiment One as well, though, of course, with appropriate simplifications.

Rules of Strict Counterpoint. The method of strict counterpoint was formalized for pedagogical purposes in the *Gradus ad Parnassum*, originally published in 1725 by J. J. Fux,¹ who systematically codified the compositional practices of Palestrina, in Fux's time the best-known composer of the highly consonant polyphonic style of the sixteenth century. Apparently, as disclosed in more recent studies,² it seems that Fux allowed some eighteenth-century practices to enter into his method. However, regardless

¹ J. J. Fux, *Steps to Parnassus* (trans. and ed. by A. Mann, with the collaboration of J. St. Edwards), W. W. Norton & Company, Inc., New York, 1943.

² K. Jeppeson, *Counterpoint* (trans. by G. Haydon), Prentice-Hall, Inc., Englewood Cliffs, N.J., 1939. See especially ix-xvi and pp. 1-53.

Table 4
Illiad Suite Experiments Summarized

Experiment One: Monody, two-part, and four-part writing

A limited selection of first-species counterpoint rules used for controlling the musical output

- (a) Monody: *cantus firmi* 3 to 12 notes in length
- (b) Two-part *cantus firmus* settings 3 to 12 notes in length
- (c) Four-part *cantus firmus* settings 3 to 12 notes in length

Experiment Two: Four-part first-species counterpoint

Counterpoint rules were added successively to random white-note music as follows:

- (a) Random white-note music
- (b) Skip-stepwise rule; no more than one successive repeat
- (c) Opening C chord; *cantus firmus* begins and ends on C; cadence on C; B-F tritone only in VII₆ chord; tritone resolves to C-E
- (d) Octave-range rule
- (e) Consonant harmonies only except for $\frac{6}{4}$ chords
- (f) Dissonant melodic intervals (seconds, sevenths, tritones) forbidden
- (g) No parallel unisons, octaves, fifths
- (h) No parallel fourths, no $\frac{6}{4}$ chords, no repeat of climax in highest voice

Experiment Three: Experimental music

Rhythm, dynamics, playing instructions, and simple chromatic writing

- (a) Basic rhythm, dynamics, and playing-instructions code
- (b) Random chromatic music
- (c) Random chromatic music combined with modified rhythm, dynamics, and playing-instructions code
- (d) Chromatic music controlled by an octave-range rule, a tritone-resolution rule, and a skip-stepwise rule
- (e) Controlled chromatic music combined with modified rhythm, dynamics, and playing-instructions code
- (f) Interval rows, tone rows, and restricted tone rows

Experiment Four: Markoff chain music

- (a) Variation of zeroth-order harmonic probability function from complete tonal restriction to "average" distribution
- (b) Variation of zeroth-order harmonic probability function from random to "average" distribution
- (c) Zeroth-order harmonic and proximity probability functions and functions combined additively
- (d) First-order harmonic and proximity probability functions and functions combined additively
- (e) Zeroth-order harmonic and proximity functions on strong and weak beats, respectively, and vice-versa
- (f) First-order harmonic and proximity functions on strong and weak beats, respectively, and vice-versa
- (g) *i*th-order harmonic function on strong beats, first-order proximity function on weak beats; extended cadence; simple closed form

of how true this is, the rules of strict counterpoint still remain a logical abstraction of many important elements of musical structure and form a fundamental basis for handling linear melodic flow.

Basically, the writing of strict counterpoint involves composing first a *cantus firmus*—a whole-note melody of up to some twelve notes in length. Against this *cantus firmus*, there are set one, two, three, and occasionally more countermelodies. Fux divided counterpoint settings into five “species,” in order to introduce compositional problems systematically. In first-species counterpoint, only note-against-note writing is permitted. In second and third species, one or more of the secondary voices moves faster, thus introducing new problems such as weak and strong rhythmic beats and passing-note dissonances. In fourth species, syncopated writing and the resolution of suspensions are treated; while in fifth-species, or florid, counterpoint, all of these techniques are combined. In our discussion, however, we shall be concerned with first species only, since only this type of counterpoint has been programmed in detail for the computer.

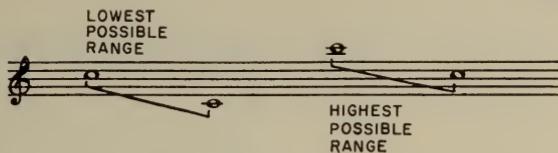
The rules derived from first-species counterpoint as used for computer programming may be grouped conveniently into three categories, namely: (1) melodic rules, (2) harmonic or vertical rules, and (3) combined rules. It may be observed that the most important departure from the authentic rules of first-species strict counterpoint is found in the treatment of the cadence. The various rules are also illustrated in Figure 2.

Melodic Rules

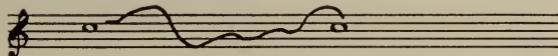
1. No melodic line may span more than an octave, i.e., the range from the lowest note to the highest note of a given melodic line should be an octave or less. The limits were set as any octave from the octave C–C' to the octave C'–C'', as shown in the first part of Figure 2.
2. If the melodic line is the *cantus firmus* itself, it must begin and end on the tonic. C was selected arbitrarily as the tonic in our experiments.
3. If the melodic line is not the *cantus firmus*, it still must begin and end on notes of the tonic chord. This was the C-major triad in our experiments.
4. A melodic skip of a major or minor seventh is forbidden because this is a dissonant melodic interval.
5. Any melodic skip, i.e., a melodic movement of a minor third or more, must be followed by a tone repeat or by a stepwise motion with or without a change of direction. A stepwise motion is a minor or major second. A stepwise melodic movement may be followed by another stepwise motion

RULE (1)

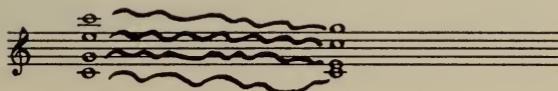
OCTAVE RANGES



RULE (2)

FIRST AND LAST NOTES
OF CANTUS FIRMIUS

RULE (3)

TYPICAL EXAMPLES OF
OPENING AND CLOSING
NOTES

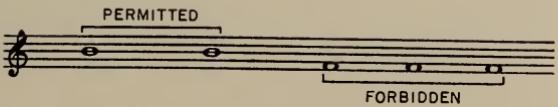
RULE (4)

TYPICAL EXAMPLES OF
FORBIDDEN SEVENTHS

RULE (5)

TYPICAL EXAMPLES OF
SKIP-STEPWISE MOTIONS

RULE (6)

FORBIDDEN THREE-NOTE
REPEATS

RULE (7)

FORBIDDEN REPEAT
OF CLIMAX

Figure 2. The rules of first-species counterpoint illustrated by means of typical musical examples.

RULE (8)

TYPICAL PERMITTED,
FORBIDDEN, AND CON-
DITIONAL INTERVALS

RULE (9)

$\frac{6}{4}$ CHORDS FORBIDDEN

RULE (10)

VII_6 PERMITTED, BUT
MUST BE RESOLVED

RULE (11)

TYPICAL EXAMPLES OF
FIRST AND LAST CHORDS

RULE (12)

TYPICAL EXAMPLE
OF CADENCE

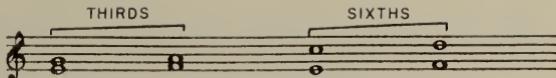
RULE (13)

FORBIDDEN PARALLEL
MOTIONS

Figure 2. (Continued.)

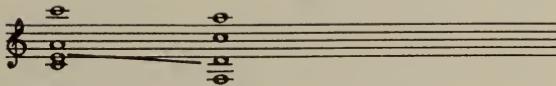
RULE (14)

PERMITTED PARALLEL MOTIONS



RULE (15)

REQUIRED STEPWISE MOTION



RULE (16)

REQUIRED CONTRARY MOTION

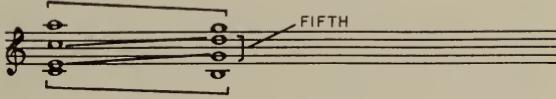


Figure 2. (Concluded.)

or by a skip. It should be noted that this skip-stepwise rule prevents the melodic line from outlining triads.

6. No more than one successive repeat of a given note is permitted.

7. It is forbidden to repeat the highest, or climax, note of a given melodic line unless it is high C and the melodic line happens to move in a tritone resolution or in the cadence up to high C. In our coding, this rule was applied only to the top voice (Violin I).

Harmonic Rules

8. Only consonant intervals are permitted. The permissible intervals include unisons, octaves, perfect fifths, and major and minor thirds and sixths. Forbidden dissonant intervals include major and minor seconds and sevenths, and the tritone, except as under Rule 10 below. By tritone, we refer throughout to both the augmented fourth and diminished fifth, which technically should be distinguished.

9. The perfect fourth is considered a consonance only if it *does not* occur between an upper voice and the voice sounding the lowest note of a chord. This would yield a triad in second inversion (a $\frac{6}{4}$ chord) which is considered dissonant. This rule automatically screens out all perfect fourths in two-part writing.

10. The tritone is considered a dissonant interval, but it is permitted if it occurs in a VII_6 chord in major (II_6 in minor). The only tritone occurring in C-major *cantus firmus* settings is the B-F interval, which is, there-

fore, permitted if D occurs below the interval. Moreover, the tritone must resolve to the minor sixth E-C, or major third C-E.

11. The first and last chords of a *cantus firmus* setting must be based on the tonic triad. This is a consequence of the second and third melodic rules. Moreover, the lowest notes of these chords must be the tonic itself. Thus, these chords must be in root position.

12. In the cadence (the last two chords of a *cantus firmus* setting) the chord just before the final chord must contain the leading tone B in any one of the four voices, but in not more than one voice. Moreover, this leading tone must move to the tonic. The other counterpoint rules also apply to the cadence. It should be observed that this is a somewhat more liberal cadence rule than that required in traditional strict counterpoint in which the next note to the last of the *cantus firmus* is required to be the supertonic D for settings in C. Actually, this more liberalized cadence formula was more difficult to program than the stricter rule would have been, but we felt that it was a more challenging problem and would produce more interesting cadences.

Combined Rules

13. Parallel unisons, parallel perfect fifths and perfect fourths, and parallel octaves are forbidden.

14. Parallel thirds and sixths are permitted, including parallel major thirds and minor sixths forbidden in the strictest counterpoint.

15. In proceeding from one chord to the next, at least one of the four voices must move by stepwise motion or remain stationary.

16. If any two voices move in the same direction into a unison, perfect fifth, perfect fourth, or octave, then one of the remaining voices must move by contrary stepwise motion or stay on the same note.

Processing These Rules for Coding. Upon examination of these rules of counterpoint, it was evident that, for setting up computer operations to be outlined by means of a block diagram, these rules could be classified as follows: (1) where voices are considered one at a time, (2) where voices are considered two at a time, (3) where voices are considered four at a time. Class 1 includes all of the melodic rules; class 2 includes all the rules designed to exclude harmonic dissonance; and class 3 includes more complicated rules such as having at least one voice moving in contrary motion in respect to others.

Secondly, it was observed that the over-all selection of notes could be divided into three classes of operation as follows: (1) initial notes, (2)

intermediate notes, and (3) cadence notes. Thirdly, while it was entirely possible to generate all the intermediate-type notes randomly and reject those which do not satisfy the rules, it was also observed that, for the sake of efficiency, it was possible to reduce the rejection rate considerably by establishing three basic subclasses of intermediate notes as follows: (1) tritone-resolution notes, (2) skip-stepwise notes, and (3) random notes. In subclass 1, the notes to be used were predetermined by the fact that a tritone would have occurred which requires a particular resolution. In such a circumstance, the generation of random notes would have been wasteful, and it was far simpler to supply the required notes directly. Subclass 2 notes were required whenever a melodic skip occurred. Here again, it would have been simply wasteful to generate random notes, and it was more efficient to restrict the note selection to the few possible choices. Since there are permitted only two possible stepwise motions plus the repeat of a note, it was convenient to generate randomly one of the three increments -1 , 0 , $+1$ equivalent to these melodic intervals and add it to the previous note. Subclass 3 notes, that is, all other intermediate notes, were generated purely randomly.

The Try-again Subroutine. Whenever a note was rejected, another note to replace it had to be generated and tested. To do this, insertion of the try-again subroutine, described in the last chapter, into the counterpoint program was required. A count was kept of the number of times the computer tried again, and if, after a sufficient number of trials, the probability of all the possible notes being tested at least once was very high and the notes were still being rejected, the computer was instructed to erase from storage the "composition" thus far completed and start all over again. As already noted in Chapter 4, the number of trials was set at fifty by a simple counting index.

A "Direction of Composition." If we define the term the direction of composition as the sequence chosen for placing notes successively in a musical composition, then we observe that a direction of composition is required the moment two-part and four-part writing is initiated. In this type of writing, in the experiments to be described, the direction of composition was set as one *harmonic line*, that is, one *vertical line*, musically, at a time. This sequence of note selection seemed to us to be a somewhat more practical and efficient method than the alternate procedure of composing initially a complete *cantus firmus* and then attempting to find other mutually compatible melodic lines which could be set against this *cantus firmus*. It should be emphasized, however, that this alternate method could

have been employed had we so desired. Therefore, the actual "composing procedure" adopted was the following for four-part counterpoint: A note of the *cantus firmus*, or Voice 1 (cello), was generated first and then followed successively by the generation of notes for Voice 2 (viola), for Voice 3 (violin II), and lastly for Voice 4 (violin I), except in special situations such as tritone resolutions and cadences in which some voices were already determined.

Indexing the Notes of the Scale. As already noted, for the generation of first-species counterpoint, it was possible to restrict the choice of notes to the white notes of the scale, since C-major *cantus firmus* settings were being considered. This was a desirable simplification, since it reduced the number of random notes which needed to be generated. Hence, efficiency was increased and the complexity of the program was reduced. The notes of the musical scale were therefore indexed in simple numerical sequences upwards from low C, omitting the sharps and flats, as shown in Figure 3.



Figure 3. Numerical index of the white notes of the musical scale as used in Experiments One and Two.

It should be mentioned that starting the index with the number 0 rather than 1 is a convenience for setting up arithmetical operations in the computer. Otherwise, this somewhat unusual method of numbering has no significance. It might also be noted that the output produced in both Experiments One and Two was printed in number notation on the Teletype machine in accord with the index shown in Figure 3. Transcription of this notation to standard musical notation was then carried out by hand.

In assigning integers in this way to represent the notes of the musical scale, it should be clear that we did not adopt some completely arbitrary procedure. In Experiments One and Two, we preserved the same order in this numerical—or stochastic—representation as that of the musical scale. Moreover, as we shall see later on in Experiment Three, when we revised this representation to apply to the ordinary chromatic scale, the numerical difference between any two of these stochastic representations corresponded

to the number of half-tone steps separating the notes being represented. Thus, numerical representations became associated with particular musical intervals. It was this fact which enabled us to add the number representing an interval to a previous note and thereby produce the correct numerical representation of a new note. With these basic decisions made, it was then possible to sketch block diagrams for each of the parts of Experiment One.

Simple Monody: Coding for *Cantus Firmi*. In this first part of Experiment One, we limited the control over the musical materials to the same rules used in the preliminary tests before the try-again subroutine was developed, namely: (1) no melodic sevenths, (2) no melodic tritones, (3) the octave-range rule, and (4) the rule that all the *cantus firmi* must start and end on C. A comparison of these few rules with the counterpoint rules listed earlier disclosed that many violations of the technique of strict counterpoint could be anticipated, and indeed, as we shall see, the results showed that all possibilities not specifically excluded by the rules employed appeared in the musical output.

The actual computer program was similar to the "melodic subroutine" to be described under Experiment Two and included in addition a simple program for fixing the first and last notes as C. The lengths of the *cantus firmi* were set by a simple counter. Thus for a *cantus firmus* containing n notes, this counter was set at $-n+1$, and after each note was selected (including the first note), unity was added to this quantity to yield the successive values $-n+2, -n+3, \dots, -n+n$. This last value, occurring after note $n-1$ had been selected, is, of course, equal numerically to zero, the lowest possible positive number recognized in computer calculations. The change in sign from a series of negative numbers to a positive number was detected by a conditional transfer order, which was then used to shift to the instructions used to terminate the *cantus firmus* with the note C and begin a new *cantus firmus*. The number of *cantus firmi* of each given length was controlled by another counter working on the same simple principle. Lastly, the lengths of the various *cantus firmi*, denoted by the function n , were obtained by means of a series of preselected stored parameters used for n .

In the first run, it was decided to generate fifty samples each of *cantus firmi* from three to twelve notes long to produce a total of five hundred of these simple melodies. The actual time required to do this in the computer was limited primarily by the time the computer required to punch the output tape.

Simple Two-part *Cantus Firmus* Settings. The next step was to consider

simple two-part writing in which some vertical relationships between the two voices had to be considered and yet in which the complications involved in crosschecking more complex relationships between more than two voices were still avoided. Therefore, for the new added voice, the same melodic rules as used for the *cantus firmus* were put in effect, except that the first notes were permitted to be E or G as well as C. Secondly, vertical dissonances, specifically seconds and sevenths, the B-F tritone, and perfect fourths, were forbidden in what was to become eventually the "harmonic subroutine" of Experiment Two. Thirdly, as a simple cadence subroutine, it was required that a leading tone B be inserted in either one of the two voices in the next-to-last chord and that this B resolve upwards to C. The selection of which voice the B was made to occur in was carried out by means of a binary random-choice operation and was also made to depend upon proper voice leading, in accord with the rules in effect. Fourthly, parallel unisons, octaves, and fifths occurring between successive chords were forbidden. The technique for accomplishing this in four-part writing is described under Experiment Two. Again, as in the case of the *cantus firmi* alone, fifty each of *cantus firmus* settings from three to twelve notes long were produced.

Simple Four-part *Cantus Firmus* Settings. In the last part of Experiment One, we extended our technique to include four voices, in accord with the primary objective of this experiment. The principal problem now became one of dealing with a set of interactions of increased complexity. In this first attempt at four-part writing, we still kept the rules relatively simple and added as few new rules of operation as possible beyond those used for two-part settings. Certain operations were actually liberalized; specifically, $\frac{6}{4}$ chords and the VII₆ chord were permitted. As it turned out, two sets of computations were completed in this last part of Experiment One. The results of the first set indicated that, even at this point, the attrition because of the restrictions already in effect was so great that certain undesirable results were occurring very frequently. Specifically, the most objectionable result was the occurrence of many repeats of a given note in the same voice. It turned out that many reiterations of one note became a convenient solution to many of the voice-leading problems which arose in this style of counterpoint. While not necessarily unattractive in sound, this was, for our purposes, an undesirable end result. Hence, for a second run, an additional rule was inserted into the program to forbid more than one successive repeat of any note.

The increase in complexity which resulted as a consequence of four-part

writing involved the problem that all the harmonic and parallel-motion rules had to be cross-checked between all voices. Thus, six separate cross checks were written into the computer instructions. These were required for the following pairs: Voice 1–Voice 2, Voice 1–Voice 3, Voice 1–Voice 4, Voice 2–Voice 3, Voice 2–Voice 4, and Voice 3–Voice 4. This became the entry of the “harmonic subroutine” of Experiment Two, which had to be entered six times in order that all the voices were properly cross-checked.

Lastly, a routine for the cadence formula, listed as Rule 12, had to be written. This turned out to be a complicated problem in itself. In this routine, described fully under Experiment Two, all the standard voice-leading and harmonic rules still had to be checked, but there were, as well, additional restrictions defining acceptable note combinations. The cadence routine was in essence a program which contained instructions embodying these additional restrictions and also a long series of rewrite orders which temporarily rewrote large sections of the regular counterpoint instructions so that they might be specifically applied to the cadence formula.

As in the previous parts of Experiment One, fifty samples each of settings from three to twelve notes long were produced.

Experiment Two

With the basic points considered under the topic of Experiment One in mind, it is now possible in discussing the details of Experiment Two to examine some block diagrams actually used in the production of musical materials in Experiment Two. It is convenient to start with the diagram for the main routine shown in Figure 4.

Main Routine for Four-part First-species Counterpoint. In the upper left-hand corner of the diagram shown in Figure 4, there is a box labeled “Initial Entry,” which represents the routine instructions carried out directly after the program is read into the computer. This block includes routine operations such as clearing storage locations, setting up indices and counters, storing parameters, and so on. It is at this point that instructions for the number of *cantus firmus* settings of various lengths to be completed in each given computer run were included in this particular program of instructions.

Initial Notes Subroutine.³ Following these preliminaries, the first im-

³ The term *subroutine* in our discussion is not restricted to the self-contained group of orders with a formalized entry and exit, sometimes more precisely called a *closed subroutine*. Both open and closed subroutines were used in this programming depending upon circumstances and convenience.

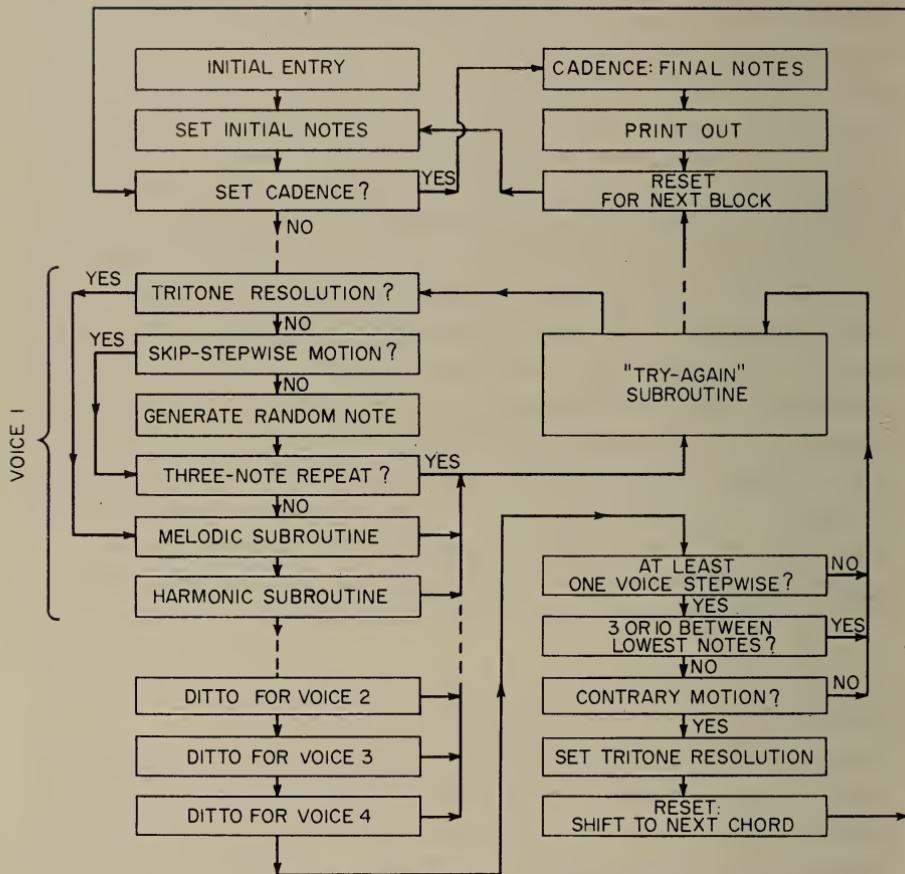


Figure 4. Experiment Two: Block diagram for the main routine.

portant task was to set the first chord of the first *cantus firmus* setting in accord with Rules 2, 3, and 11, given on pages 84 and 88. This was done by means of a special subroutine required only at the beginning of each *cantus firmus* setting. Once it had been used, this subroutine was not entered again until after a setting had been completed and the computer was ready to proceed with the generation of the next setting.

The basic problem was to set the chord of C major *in root position* into the four voices in the most efficient manner. Since only the notes C, E, G, C', E', G', and C'', with numerical equivalents of 0, 2, 4, 7, 9, 11, and 14, respectively, were involved, it was possible to restrict the choice of random integers to these few numbers by storing these particular numbers in a special small table of seven entries and selecting the memory locations

for these numbers by random-integer generation, utilizing the technique already described in Chapter 4. Moreover, since the initial note for the *cantus firmus* (Voice 1) was limited even more to just C, C', or C'', to make the note-selection process as simple as possible, we could arbitrarily select C', for which the numerical equivalent is 7, for this voice. This automatically ensured that the chord would be in root position, since Voice 1, being for the cello, would subsequently be transposed downward an octave.

Main Routine Continued. Since *cantus firmus* settings from as short as three notes to any reasonable length might be required, the next logical step was to ask whether the next-to-last chord had been reached and whether the cadence routine would be required. This would happen immediately, of course, only when the length of the setting had been restricted to three notes. However, it was desirable that this decision be made at this point because of the cyclic nature of the program. In the experiments actually carried out with the Illiac, the longest settings generated were twelve notes long, and all the output for Experiment Two, in particular, was limited to this length. However, this was not the case in Experiment One, as we have already seen. This decision operation, which not only brought the special cadence routine into operation, but also set the lengths of the *cantus firmus* settings, worked in the following way: If we wished to write settings for *cantus firmi n* notes long, a cadence routine would be required after $n - 2$ notes had been generated in each voice, since both the last and next-to-last chords would be involved in the cadence. Therefore, the number $-(n - 2)$ could be set in a convenient storage location, and after each chord was successfully completed, a unit increment could be added to this quantity to obtain successively the values $-(n - 3)$, $-(n - 4)$, . . . , -1 , 0, as shown in Table 5. Again since zero is considered a positive number, when the operation $-(n - 2) + (n - 2)$ was carried out after $n - 2$ chords had been formed, a change of sign took place. This change of sign activated the conditional transfer order and caused the sequence of operations to shift to the special cadence routine. Until this change of sign occurred, however, the transfer order was bypassed and the principal part of the program, which was used for the generation of intermediate notes, was entered.

The next group of six blocks of instructions bracketed as referring to Voice 1 was repeated with minor variations also for Voices 2, 3, and 4, taken successively. These were blocks of instructions primarily derived from rules for the voices taken one at a time or two at a time. In fact, these

Table 5
Experiment Two
Counting Operation for Activating the Cadence Routine

Last chord formed	Index plus counter	Effect on conditional transfer order to cadence
Initial chord	$-(n - 2)$	$= -(n - 2) + 0$ $= -(n - 2)$
Second chord	$-(n - 2) + 1$	$= -(n - 2) + 1$ $= -(n - 3)$
Third chord	$-(n - 3) + 1$	$= -(n - 2) + 2$ $= -(n - 4)$
	$-(n - 4) + 1$	$= -(n - 2) + 3$ $= -(n - 5)$
	.	.
	.	.
	.	.
$(n - 3)$ rd chord	$-(n - n + 2) + 1 = -(n - 2) + (n - 3) = -1$	<i>Positive number</i>
$(n - 2)$ nd chord	$-(n - n + 1) + 1 = -(n - 2) + (n - 2) = 0$	<i>Transfer to cadence routine</i>

operations took care of all of the rules for intermediate notes except some of the more complex interactions which could only be tested for after all four notes were selected.

Within the group of instructions for each voice, the two special types of notes which could be set without resorting to the general random note scheme were treated first. In the set of more general instructions—on the right-hand side of Figure 4—the block labeled “Set Tritone Resolution” is a set of instructions which was activated whenever a tritone—which in these experiments was limited to the combinations F-B, F-B’, B-F’, F’-B’—had occurred between any two voices in the previous chord. The “Set Tritone Resolution” operation also recorded between which voices the tritone had occurred and stored this information for the “Tritone Resolution” block for Voices 1, 2, 3, and 4. Lastly, it also eliminated tritones occurring between more than two voices, which could arise only as a consequence of doubling at the unison or octave. These doublings were unacceptable, since they would produce forbidden parallel motions at the unison or octave. It should also be noted that before the tritone-resolution notes were set for the next chord, they were examined by means of the melodic subroutine to be described below. If they were found to be unacceptable, the chord was erased and started over.

The tritone-resolution operation was used therefore, whenever a tritone had occurred between two voices only, and only when one of the tritone-interval notes had occurred in the voice for which a new note was being generated. If the tritone note in the previous chord happened to have been F or F’ (index numbers 3 or 10) or B or B’ (index numbers 6 or 13), the notes in the new chord were automatically inserted as E or E’ (2 or 9) or C or C’ (7 or 14), respectively. Otherwise, the operation was bypassed.

The skip-stepwise operation was based upon Rule 3 and was set up so as to be more efficient than testing randomly generated notes for Rule 3. The two notes previous to the one being generated were first examined by finding the negative absolute value of the difference of their values. Thus, if we were about to generate note N_i , we computed $-|N_{i-1} - N_{i-2}|$. The quantity 1 was then added to this value, and if the result turned out to be positive, we knew that the previous melodic interval had been smaller than a third, that is, it had been a stepwise motion. The positive sign, if it occurred, was used to activate the conditional transfer order, this time to bypass the special skip-stepwise routine and to proceed to the generation of a random note. On the other hand, if the number was found to be negative, we simply generated at random the values +1, 0, or -1 for the

interval $N_i - N_{i-1}$, bypassed the general random-note-generation process, and proceeded to further testing. This calculation is illustrated in Table 6.

Table 6
Experiment Two
An Illustration of the Process for Detecting Skips in
Applying the Skip-stepwise Rule

Note N_{i-1}	Note N_{i-2}	$- N_{i-1} - N_{i-2} $	$- N_{i-1} - N_{i-2} + 1$
C'(7)	C'(7)	0	+1 } Stepwise motion
A(5)	B(6)	-1	0 } Generate random note
E'(9)	G'(11)	-2	-1 } Skip
C''(14)	G'(11)	-3	-2 } Generate +1, 0, or -1 etc. . for the interval, . $N_i - N_{i-1}$. .

To generate the increments +1, 0, and -1, we simply used the random-integer-generation process previously described in Chapter 4, setting $b = 3$ to yield the values 0, 1, and 2 in random sequence. By subtracting 1 from these values, we obtained in turn -1, 0, and +1 in random sequence.

The next step in the process was the generation of a random note without restriction. Since there were fifteen possible notes available for selection, n was set at 17, this being the smallest prime number above 15, and the integers 15 and 16, whenever they turned up, were rejected by means of the process of the type shown in Table 7.

The next step was the elimination of multiple repeats of the same note, specifically, the elimination of all repeats of the same note but one. To do this, the quantity $|N_i - N_{i-1}|$ was first computed, and if it was found to be unequal to zero, the test was concluded, since this meant that the previous melodic interval had *not* been a repeat. On the other hand, if the difference was found to be zero, then further testing for the second interval back was also required. The interval $|N_{i-1} - N_{i-2}|$ was then computed. If this was found to be unequal to zero, we know the second melodic interval back had *not* been a repeat. In this case, the test was again concluded. However, if this difference was also found to be zero, then two repeats in a row had occurred. Since this is forbidden by the rules, the generated note, in this case, had to be rejected and try-again subroutine entered.

Melodic Subroutine. The melodic subroutine, which in simplified form,

but with additions for selecting the first and last notes of a *cantus firmus*, had been also the first music-generation code written for the Illiac as part

Table 7
Experiment Two
Process for Rejecting Certain Random Integers
(Specifically 15 and 16) Out of the Set, 0, 1, . . . , 16

Random integer	Random integer — 15	Conditional transfer result
0	—15 }	Accept
1	—14 }	
.	.	
.	.	
.	.	
.	.	
.	.	
14	—1 }	
15	0 }	Reject and recycle to repeat
16	+1 }	random-integer generation

of Experiment One, was used both to screen out forbidden intervals between successive notes and to check longer-range interactions between the notes of a given melodic line.

In the testing of intervals between successive notes, a simple additive process was employed. Initially, $|N_i - N_{i-1}|$ was computed and the quantity 7 subtracted from this absolute difference. As shown in Figure 5, each time we tested this yielded one of a series of integers which could then be checked to find the exact magnitude of the melodic interval. Since the only forbidden melodic intervals were sevenths and tritones, only these had to be screened out. Therefore, it was possible to test immediately for whether the interval was an octave or larger, and if so, the interval was conditionally accepted. Since intervals larger than an octave were automatically eliminated by the test to follow for the octave-range rule, the octave was, in effect, being detected by this first screening operation. Directly thereafter, it was possible to test for the seventh. If the seventh was found, the try-again routine was entered, and the whole process started over again. On the other hand, if the seventh was not found, then, by the rules, all the remaining intervals were acceptable except the tritone, and the possibility of the presence of this interval could immediately be tested for. Therefore, a screening operation for sixths could be omitted.

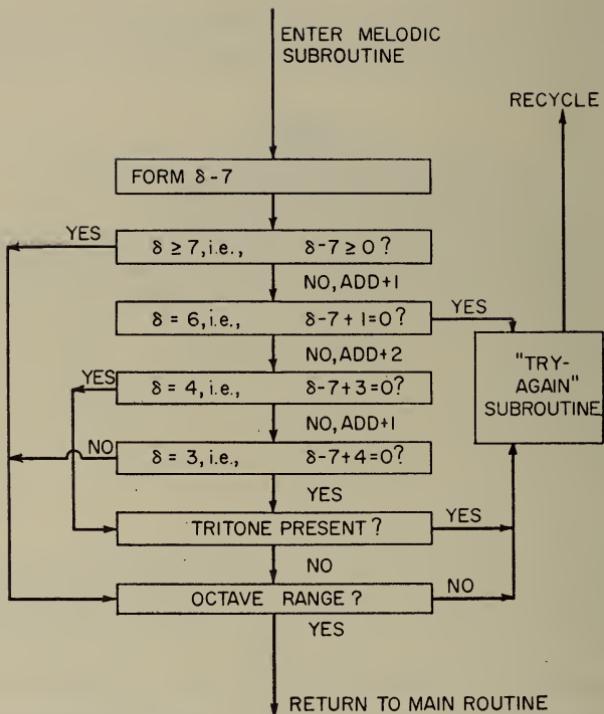


Figure 5. Experiment Two: Block diagram for the melodic subroutine.

It was possible to proceed directly to the detection of the presence of fifths and fourths, that is, of whether $\delta = |N_i - N_{i-1}| = 4$ or 3. If one or the other of these intervals was found, the presence of a tritone was possibly indicated and additional screening was required. Thus, if the difference $\delta = 4$ was found, then a tritone involving the notes B-F' (indexed as 6 and 10) had to be tested for by a subtractive process. Alternatively, if the difference $\delta = 3$ was found, the possibility of the existence of one of the other two different tritone intervals, F-B and F'-B', had to be checked. Again, subtractive tests were employed for this purpose. Moreover, the higher interval F'-B' was first lowered by an octave, so that the same test could be employed for both these intervals.

The next operation involved the octave-range rule, Rule 1, which was, for us, one of the more interesting rules of counterpoint. As noted previously, this rule states that a melodic line is limited to a span of an octave or less, but it does not specify which particular octave span this range must encompass. This operation is in some ways more generalized

than most of the others, for the specific limitations it imposes upon the melody develop only as the melody is being written. Therefore, each time a note was generated, track had to be kept of whether this new note was more than an octave higher or lower than *any other note* already present in the melody. If not, then the note was provisionally accepted and the test concluded. On the other hand, if the note violated this rule by exceeding an octave, then the try-again subroutine was entered. This test operated quite simply as follows: The note just generated, N_i , was subtracted from note N_1 . The quantity 8, which is equivalent to the span of a ninth, was then subtracted from the absolute value of this difference, and if the result remained positive, we knew that a skip larger than an octave between this most recent note and the first note of the melodic line had been formed and the note N_i was rejected. This process was then repeated successively for $N_i - N_2$, $N_i - N_3$, to $N_i - N_{i-1}$, and only if the condition required by Rule 1 was fulfilled for all these intervals was the note accepted.

Finally, the repeat of a climax in the melodic line of the highest voice, which, in accord with Rule 7, we considered to be forbidden, was tested for. Although this routine was not a part of the melodic subroutine since it had to be employed only in connection with the highest voice, it is convenient to mention it at this point. As each note in the highest voice was accepted, a record was kept of the highest note generated up to that point, and a specific test was eventually made of whether a repeat of this note had occurred. This rule, however, was not applied in the cadence or in conflict with the tritone-resolution rule.

Harmonic Subroutine. The principal function of this subroutine was to screen out vertical dissonances and to restrict chords to perfect triads and their first inversions, the one chord with a permissible dissonance being VII₆, which contains the tritone interval. The block diagram for this subroutine is shown in Figure 6. In Figure 6, we have used the symbol Δ to represent a vertical interval, that is, the absolute numerical difference between the notes in two different voices on the same beat.

The subroutine has been simplified in Figure 6 to show how the testing is carried out between any two voices, but it should be remembered that the relationships between all possible pairs of voices had to be examined. The six possible combinations of voices taken two at a time are V_1-V_2 , V_1-V_3 , V_1-V_4 , V_2-V_3 , V_2-V_4 , and V_3-V_4 . The harmonic subroutine had, therefore, to be entered a total of six times before a particular combination of notes could be passed as acceptable. It was most efficient to design a

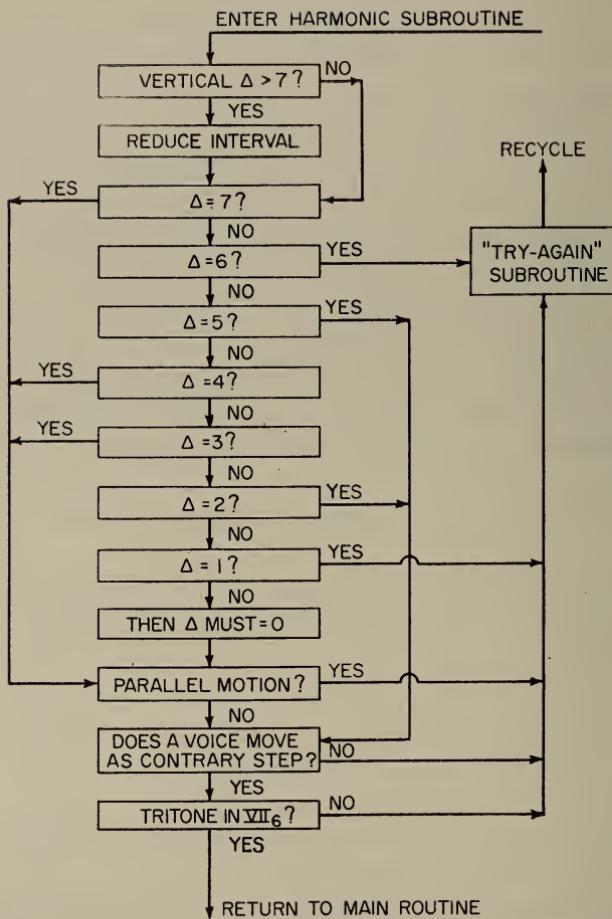


Figure 6. Experiment Two: Block diagram for the harmonic subroutine.

recycling operation for testing all four voices by means of the melodic and harmonic subroutines by using a set of indices which would keep track of what was tested and which would also carry out the testing in the most efficient manner. The order of testing is shown in Table 8.

The harmonic subroutine was used (1) to eliminate vertical sevenths and seconds, (2) to eliminate parallel unisons, fourths, fifths, and octaves, all of which are forbidden by the rules, (3) to test whether a unison, fourth, fifth, or octave had been formed by the two voices moving in the same direction by contrary stepwise motion, or by one or both voices remaining stationary, and (4) to test whether a tritone had occurred. The testing of

Table 8
Order of Testing the Four Voices with the
Melodic and Harmonic Subroutines

Voice 1

Enter melodic subroutine for V_1

Harmonic subroutine is not entered since there is nothing to test

Voice 2

Enter melodic subroutine for V_2

Enter harmonic subroutine for $V_1 - V_2$

Voice 3

Enter melodic subroutine for V_3

Enter harmonic subroutine for $V_1 - V_3, V_2 - V_3$

Voice 4

Test for repeat of climax note

Enter melodic subroutine for V_4

Enter harmonic subroutine for $V_1 - V_4, V_2 - V_4, V_3 - V_4$

whether D, rather than B or F, was the lowest note of the one chord which might contain a tritone, that is, of whether the chord was VII_6 and not VII or VII_4 , was done later in the main routine after the whole chord being formed had been completed.

Table 9
Experiment Two
Reducing Vertical Interval to an Octave or Less

Interval	Δ	$\Delta - 8$	Reduced Δ	Δ for testing
Double octave	14	6	7	0
Fourteenth	13	5	6	-1
.
.
.
.
.
Tenth	9	1	2	-5
Ninth	8	0	1	-6
Octave	7	-1	7	0
Seventh	6	-2	6	-1
.
.
.
Unison	0	-8	0	-7

In order to simplify the testing procedure, all vertical intervals larger than an octave were first reduced. This was accomplished by computing the absolute difference $\Delta = |V_a - V_b|$ and subtracting 8 from the result. If a positive number resulted, we knew from this calculation that the original interval was larger than an octave, and +1 was added to the result to yield the reduced interval. On the other hand, if a negative answer was obtained, the original interval was employed directly for the tests. This calculation is shown in Table 9.

The next step was to eliminate sevenths and seconds and to set up the remaining tests in the subroutine. For convenience in applying the conditional transfer order, 7 was subtracted from these adjusted values of Δ just computed. Sevenths and seconds were first eliminated by adding unity successively to $\Delta - 7$ and, after each addition, testing with the conditional transfer order in order to place the interval into the proper category for subsequent processing. The technique for doing this is shown in Table 10.

In the parallel-motion test, which was applied only when $\Delta = 0, 3, 4$, or 7 , Δ_i was compared with Δ_{i-1} to see whether $\Delta_i = \Delta_{i-1}$. Whenever this occurred, we knew forbidden parallel motion had occurred, either actually, as in the examples shown in Figure 7a, or implicitly by octave

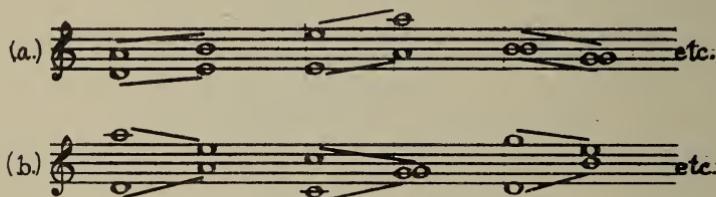
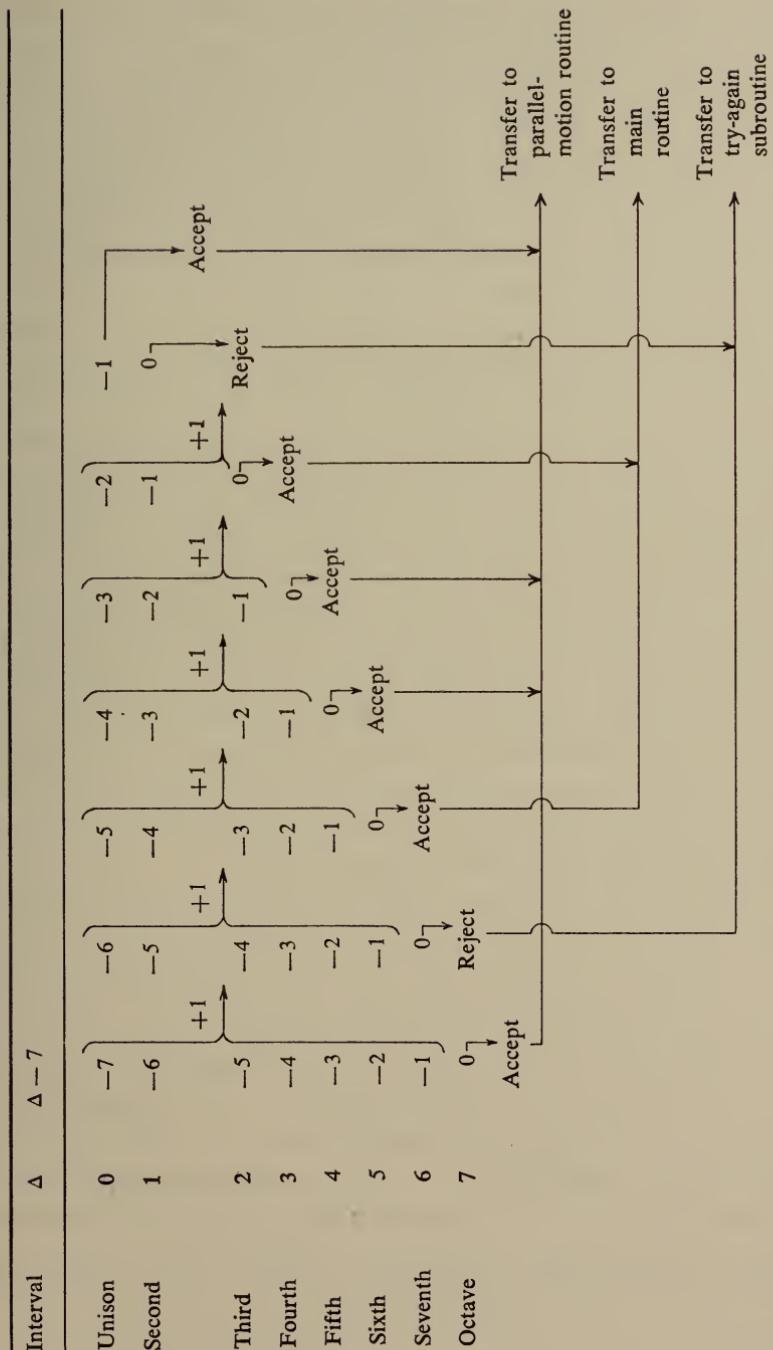


Figure 7. Experiment Two: Examples of (a) direct parallel motion, (b) implicit parallel motion.

displacement, as in the examples shown in Figure 7b. The previous interval for the two voices in question, Δ_{i-1} , was obtained from storage and reduced if necessary. It was then directly tested by subtracting from it the quantity Δ_i . If the difference was found to be unequal to zero, we knew immediately that parallel motion had not occurred.

The two remaining tests included with the harmonic subroutine also depended on the examination of the intervals between the voices taken two at a time. In the test for contrary motion, again only applied whenever $\Delta = 0, 3, 4$, or 7 , the signs of the differences $N_i - N_{i-1}$ for each of the two voices in question were determined. If the signs were unlike, then con-

Table 10
Experiment Two
Calculations for Testing Vertical Intervals between Pairs of Voices



trary motion had occurred and the test was concluded, since contrary motion was permissible. But if like signs were found, the two voices had to be tested for whether one or both of them had remained stationary, that is, for whether the value $N_i - N_{i-1} = 0$, which was also permissible as such. Otherwise, it was recorded that a possible violation of Rule 16 was indicated and that further testing in the main routine would be required to determine whether one of the remaining voices had concurrently moved by contrary motion. The testing for sign was done directly by means of the conditional transfer order.

In the last test, the specific reduced interval F-B was first searched for whenever Δ had been found by previous testing to be equal to 3 or 4. This was done by determining whether any of the quantities $N_i - 3$, $N_i - 6$, $N_i - 10$, or $N_i - 13$ equaled zero in each of the two voices to be tested. If so, it was noted that a tritone had occurred, and the rest of the test was then carried out. In the second part of this test, the presence of doubling either at the unison or octave was checked, and if found, the doubling was removed by erasing the chord and starting over again by means of the try-again subroutine. Finally, in a later part of this test, the presence in one of the remaining voices of a D lower than the tritone interval was tested for by first finding the lowest note of the tritone interval and then subtracting from this, successively, the notes in the two remaining voices. If the difference was found to be positive, we knew that a note lower than the interval existed, and we also knew that this note had to be a D, since the notes C and A form dissonant intervals with B, and the notes E and G form dissonant intervals with F.

Main Routine Continued. Once all the tests embodied in the melodic and harmonic subroutines had been successfully passed, the main routine was returned to in order that the notes now provisionally accepted could be screened through tests expressing the rules related to combined interactions between the voices. As shown in Figure 4, there were three tests which had to be satisfied. The first, based on Rule 15, required that at least one voice out of the four be required to move stepwise. This rule was checked rather easily since all that was necessary was that at least one voice be found for which $|N_i - N_{i-1}| = 1$. As soon as this condition was found to be satisfied, the test was concluded and the next operation started. This was the process needed to eliminate $\frac{6}{4}$ chords. The procedure used was the following: The lowest note of the chord was first found by testing whether Voice 1 might happen to have it by subtracting this note from the

notes of the chord in Voice 2, Voice 3, and Voice 4, in that order. This process was kept up as long as the difference remained positive. If all these differences did remain positive, it was evident that Voice 1 contained the lowest note of the chord, and this part of the test was concluded. On the other hand, if the difference in any of these subtractions turned out to be negative, we knew that the other voice of the pair being examined contained the lower note, and it was used to replace Voice 1 in subsequent screening for the lowest note. This procedure was repeated until the lowest note was found. The next step after this was to set up a check for whether a difference of 3 or 10 existed between this lowest note and any other of the three other notes. If not, then we knew that a $\frac{6}{4}$ chord had not been formed, and we were able to proceed to the next test, which was the check for contrary motion in at least one voice whenever a unison, perfect fourth, fifth, or octave formed by motion in the same direction had been detected in the harmonic subroutine as described above. Two memory locations were cleared to zero—one for positive increment and one for negative increment for the four melodic intervals between the chord under examination and the previous chord. The four melodic intervals were then computed one at a time. If a given melodic interval was found to be zero, i.e., if it was found to be a melodic repeat, nothing was entered into the two memory locations. On the other hand, if the interval was found to be positive or negative, the appropriate one of the two aforementioned locations was made nonzero. After all four intervals had been processed, both memory locations were tested for nonzero value. The actual magnitudes contained therein did not matter; the important thing to determine was whether only one of the two locations was nonzero. If this condition was found, then no contrary motion had occurred and a new chord had to be generated. On the other hand, if both were nonzero, this showed that the requirement for contrary motion had been satisfied, and we could then immediately proceed to the next operation, which was the setting up of the tritone-resolution process, described earlier, for chords containing a tritone interval.

Lastly, the two "Reset" operations, one for shifting to the next chord of a given *cantus firmus* setting and the other for starting a new setting, were both basically clearing operations for resetting indices back to original values. In addition, the "Reset: Shift to Next Chord" operation also carried out the process of storing the chord formed into the computer memory until the time when it was to be printed out or, alternatively, to be erased as a consequence of the try-again subroutine.

Cadence Routine. As already noted in our discussion of Experiment One, a routine had to be written for the cadence whenever chord $n-2$ was reached, the length of the *cantus firmus* setting being set at n notes. Moreover, in this routine, as observed before, all the standard voice-leading and harmonic rules still had to be checked. In addition, further restrictions embodied in Rules 2, 3, 7, 11, and, in particular, Rule 12 had to be recognized and properly tested for.

There were two basic procedures which were open to us. The first was to write a complete new set of orders which would test for all the rules applicable to the cadence as a self-contained closed subroutine, while the second was to assemble a set of overwrite orders so that we might apply the main routine for the major portions of the *cantus firmus* setting also to the cadence. This latter procedure would require some additional testing procedures for the extra restrictions required by the cadence formula but would eliminate duplication of various standard tests already discussed. For this latter reason, this second procedure was adopted as the preferred one.

In the cadence routine as written, the first step was to place the note B (or B') in one of the four voices. Two operations were required here. The first was to select either B or B' by two-choice random-integer generation, while the second was to determine by four-choice random-integer generation into which of the four voices this note was to be inserted. The operations required to check the rules of melodic writing contained in the main routine were then called into play to determine whether the selected note satisfied these rules. If so, the remaining notes of the chord were selected by the ordinary note-selection processes already described and screened for the usual restrictions. In actual fact, this eliminated all chords but V, V_6 , III, III_6 , and VII_6 . In addition, a new test had also to be included to eliminate doubling of the leading tone B, either at the unison or octave. This test was carried out by means of simple subtractive tests of the general type already described.

Once the leading tone chord at position $n-1$ was obtained, the final chord could be selected. Since this had to be a tonic chord in root position just like the initial chord, the subroutine for selecting the initial chord could readily be adapted to this purpose. It was modified, however, to ensure that the leading tone progressed to C, and following this, the remaining notes were generated until an acceptable combination was found. In so doing, the restriction as to the repeat of the highest note of the highest voice

was relaxed to the extent that a repeat of C'' occurring on this last chord was permitted if the preceding note happened to have been B'.

The extensive set of rewrite orders required for this cadence subroutine involved the adapting of test procedures from the other parts of the computer program for Experiment Two. A complicated bookkeeping operation was needed to set up the testing procedures for the cadence and to reconstruct the tests in their original form for the next *cantus firmus* setting after the cadence had been selected. The details of these instructions need not be considered here since, although complex, they were entirely routine in nature.

Printout. Printout instructions were activated in the program for Experiment Two whenever a *cantus firmus* setting was completed. The setting was read from the memory and printed by means of standard subroutines available for the Illiac. After printing (actually, after output tape was punched), the memory locations used for storage of the *cantus firmus* setting were cleared, and the resetting of the machine for writing the next sample of counterpoint was initiated. A sample of computer output produced by means of this program is shown in Figure 8.

07	08	05	06	09	09	12	11	09	09	11	09
09	06	07	04	05	02	03	06	05	05	02	09
09	13	12	09	09	12	12	11	12	14	11	11
07	11	12	06	05	09	08	11	12	07	06	07

Figure 8. Experiment Two: Sample of typical printed output in which some counterpoint rules are in effect.

Varying the Number of Rules Used. As noted in the Outline of Experiments and as shown in Table 4, musical output was generated by means of this program to include examples of *cantus firmus* settings in which the number of restrictions were varied from purely random white-note writing, in which no rules whatsoever were applied, to the most restricted writing, in which the full set of instructions was utilized. The various types of settings between these two extremes produced in a series of computer runs are tabulated in Table 4. Actually, the experiments were carried out in reverse order. The most complicated program, the one we have just described, was written first. After this was in working order and producing output, it was then a simple task to reduce its complexity by means of over-

write orders which inserted bypass, or "unconditional transfer orders," into the program in front of the tests to be eliminated. When a sufficient number of these bypass orders had been inserted, the program was reduced to the simple process of generating random white-note music.

In conclusion, it is of interest to note that this was the most complex single music-generation program we have written so far for the Illiac. The total number of individual arithmetical instructions required by this program for writing strict counterpoint exceeded 1,900 individual operations. In this respect, this program forms a considerable contrast to some of the codes to be discussed in connection with Experiment Three and Experiment Four, in which musical output of a less restricted nature was studied.

Experiment Three

Since the technical problems explored in this third section of work were rather varied, it is convenient to consider each separately and then describe how the various musical elements studied in this experiment were combined.

Rhythm. Rhythm was perhaps the most important musical element we felt had to be treated if a fundamental compositional technique utilizing computers was to be developed. Our objective in considering rhythm as a musical entity to be treated by computer processing was in accord with a recognition of this condition. It was our purpose to write a practical computer program for generating rhythms so that a fundamental technique might be demonstrated which in turn could form a basis for the further elaboration of rhythmic devices in more complex contexts.

Just as the notes of the scale can be represented by numbers in the computer, so also is it possible to symbolize rhythms numerically. In particular, the binary representation of numbers in the machine, consisting of permutations of ones and zeros, offered a convenient set of symbols with which we could operate. Thus, if we let 1 represent the "sounding," or "strike," of a note and 0 a "rest," or, alternatively, the "hold" of a previously sounded note, then we can let a sequence of ones and zeros represent a sequence of note values. For such sequences, ones and zeros, being integers, would each represent some arbitrarily chosen unit time interval, such as a quarter note, eighth note, or some smaller value. We next observe that a rhythmic pattern is generated by the interaction of these note values with meter, which is represented in turn by metrical patterns such as

$\frac{4}{4}$, $\frac{3}{4}$, $\frac{5}{8}$, or $\frac{6}{8}$. Thus, the production of a rhythmic structure not only in-

volves the choice of note values, but also the choice between a repetitive or a varying meter and, moreover, the selection of which particular meter or meters to use. However, in order to keep the problem relatively simple in these first experiments, we arbitrarily restricted our studies to a simple metrical pattern, namely, $\frac{4}{8}$ meter, consisting of four eighth-notes to the measure. By further restricting the choice process to eighth notes as the smallest possible time intervals, we could then utilize simple permutations of four binary digits to represent the four beats of a measure in $\frac{4}{8}$ time. There are sixteen possible permutations of these four binary digits, permutations which also happen to represent the decimal numbers 0 through 15. These could be used to represent the possible rhythms in a $\frac{4}{8}$ measure,

as shown in Table 11. Consequently, it was possible to propagate random integers between 0 and 15 and to let the values of these integers set the required rhythmic pattern for each measure. It might be noted at this point that rhythm in other metrical systems can also be generated by entirely similar processes. For triple meter, we generate random integers between 0 and 7, and for quintuple meter, random integers between 0 and 63.

Once the basic scheme for generating rhythms was set up, it was observed that, like note propagation, rhythm also is normally subjected to rules of composition and not permitted to occur entirely at random, except for the extreme case of random music, for which an appropriate random-integer table would have to be used. A table for this purpose would be based upon a scale of integers sufficiently extended so that the smallest useful time increment might be generated. It is to be noted that our rhythm-generation scheme, even though it can hardly be considered subject to much restriction, nevertheless represents a significant departure from this most extreme situation since no time intervals smaller than eighth notes were produced by it. Aside from this, however, if it were used as is, it would still generate a new rhythmic pattern in each voice in every measure produced. But rather than carry out this rather rudimentary process, it seemed desirable to proceed to something somewhat more challenging and to consider factors which might yield some rhythmic order over and above this relatively primitive situation. Since the simplest form of rhythmic redundancy is literal repetition, the first step we utilized to reduce further the randomness of the rhythm was a simple random repetition scheme. In each voice, a rhythmic pattern was generated according

Table 11
Basic Rhythmic Scheme for $\frac{4}{8}$ Meter

DECIMAL NUMBER	BINARY NUMBER	RHYTHMS	
		CLOSED	OPEN
0	0000		
1	0001		
2	0010		
3	0011		
4	0100		
5	0101		
6	0110		
7	0111		
8	1000		
9	1001		
10	1010		
11	1011		
12	1100		
13	1101		
14	1110		
15	1111		

to the method outlined above and then a subsidiary random integer, which was permitted in our particular experiment to have values between 1 and 12, was also generated. This subsidiary parameter was used to control the number of measures a particular rhythmic pattern would be sustained before a new rhythmic pattern was generated for the voice in question. In this way, we obtained a variety of rhythmic patterns sustained for different lengths of time in the different voices. This was the first and simplest of the rhythm codes we produced.

It is a common characteristic of music, however, that there exist correlations of rhythms between the voices of a polyphonic piece of music. In the above rhythmic code, this does not happen except by pure coincidence. Therefore, the next step we considered was the introduction of some vertical as well as horizontal rhythmic redundancy. In a second rhythm code written for the Illiac, we included the generation of still a second subsidiary random integer, this random integer taking values from 0 through 15. In this code, use was made once again of binary notation as a type of musical symbolism. This time, in this new subsidiary random number, ones were used to represent voices required to play the same rhythm for the number of measures determined by the first subsidiary random integer, the lower voices being used as the master voices, while zeros were used to represent voices for which rhythms were generated independently. Thus, the representation 0000 meant that all voices would have independently generated rhythms, so that vertical duplications would occur only by coincidence. On the other hand, the representation 1111 was used to mean that all voices would have to play the same rhythm, the rhythm being that generated for the lowest voice. In between these two extremes, a representation such as 0101, for example, might be generated. This particular binary number indicated that Voices 2 and 4 would play the rhythm generated for Voice 2, while Voices 1 and 3 would be rhythmically free. It should be noted, however, that this redundancy scheme could not be rigidly applied because the integers 0001, 0010, 0100, and 1000 were meaningless in this application, i.e., these integers would have represented three voices free and one the same (as what?). Also, we could not represent the cases of two different pairs being the same separately, i.e., 1-2, 3-4; 1-3, 2-4; and 1-4, 2-3. Three of the above meaningless integers were therefore appropriated for these cases, leaving one integer which was disregarded. Lastly, the complication which arose the moment these two types of rhythmic redundancy were combined, namely,

which would take precedence, was settled by letting the vertical-combinations rule take precedence.

The problem of notation of rhythm on the Teletype printout using ordinary symbols was handled by means of a makeshift code, utilizing a block of four digits to represent the rhythm in a given measure. We simply employed the binary number notation shown in Table 11 and printed twelve blocks, each consisting of four rows of four digits each, to represent the rhythm in twelve measures of four-part writing. An example of this will be illustrated in combination with other notation to be described.

This rhythm code was also arranged so that a new combination of rhythms would be generated at the beginning of each new line. There was no compelling reason for this; it was simply convenient to do so. This might also be regarded as one way of blocking in an extremely elementary longer-range structure for rhythmic patterns.

Dynamics. A second problem of considerable musical importance is the setting of dynamics. In Experiments One and Two, during the transcription of the experimental results to score form, an extremely simple scheme involving a four-integer random-integer table was employed to assign arbitrary dynamics marks to the individual voices (see Chapter 6). It was decided, however, to effect some improvement over this elementary technique, and not only place the generation of dynamics marks under the control of a computer program, but also include *crescendi* and *diminuendi* so that more than just terrace dynamics would occur.

This problem, in itself, is quite simple, and a set of operations similar to the rhythm scheme just described was worked out. The dynamic range was limited to *pp*, *p*, *mp*, *mf*, *f*, and *ff*, that is, to six possible values, so that the first operation became the simple process of generating a random integer considered equivalent to one of these dynamics marks. Secondly, a change of dynamics was indicated by means of a second random integer obtained from the possible choices 0, 1, or 2, which were considered equivalent to *diminuendo*, no change, and *crescendo*. In choosing this second random integer, restrictions had to be placed on the choice if *pp* or *ff*, the limits of the range, had been chosen previously, since *pp* could not be followed by *diminuendo*, and *ff* by *crescendo*. Therefore, a simple screening operation was inserted at this point to eliminate these two unacceptable possibilities.

Thirdly, the duration of the dynamics indication had to be set. Since this program was written in conjunction with the rhythm code just described, techniques similar to those used for the rhythm code were em-

ployed. In the simpler and earlier of the two codes actually written, a random integer which could take the values 0, 1, . . . , 12 was employed to set the dynamics scheme in each voice independently. In the second revised code, vertical correlation was used, just as in the rhythm code, so that all possible combinations from complete independence to complete agreement might be achieved.

The dynamics code was not correlated to the rhythm code. An entirely independent set of parameters governed the two operations, even though both were contained in the same computer program. Thus, the dynamic markings in Experiment Three are at no time correlated to rhythmic patterns or to changes in rhythm. Obviously, a correlation of one type or another could have been set up had we so desired, but for the sake of simplicity, this was not done.

Since letter notation is commonly employed to indicate dynamics, the printing of dynamic indications offered little difficulty. The letters F, P, and M were reserved for this purpose. To indicate *crescendo*, the symbol, (, was appropriated; to indicate constant dynamic level, the symbol, =, was used; while the symbol,), was used for *diminuendo*.

Orchestration Index. This is a name we have applied to the playing instructions which might be used to give color or timbre variation to the sound of the music being produced. In the most general sense, this can be thought of as the basis for a technique of instrumentation. In our experiments, since we planned the output for string quartet performance and thus had already made the choice of the particular instruments to be used, this problem resolved to the question of the various ways of playing stringed instruments. As is well known, there are numerous ways stringed instruments can be played, including various manners of bowing, several types of *pizzicato*, and a number of more sophisticated techniques, such as *col legno* (bowing with the wooden part of the bow), *sul ponticello* (playing on the bridge of the instrument), the playing of harmonics by causing the strings to vibrate in sections, and so on. In order to prepare a set of playing instructions, we selected sixteen of the most common stringed-instrument playing techniques and tabulated these as shown in Table 12. With the bowing instructions, we also included indications for transcription purposes as to whether a zero in the rhythmic code was to be observed as a sustained note or as a rest. These instructions for sustaining a note or for observing a rest were associated, as a general rule, with the bowing instructions with which they were most conveniently paired. Lastly, to allow for the fact that impossible combinations perform-

Table 12
Experiment Three
Orchestration Index

Random sexidecimal integer	Playing instructions	If playing instructions are impossible, revert to
0	Bowed <i>legato</i> , held through rests	F
1	Bowed <i>detaché</i> , rests observed	F
2	Bowed <i>tremolo</i> , hold through rests	0
3	Bowed <i>sul ponticello</i> , rests observed	1
4	Bowed, artificial harmonics, hold through rests	0
5	Bowed <i>col legno</i> , rests observed	1
6	Bowed <i>sul tasto</i> , hold through rests	0
7	Bowed <i>martellato</i> , rests observed	1
8	Bowed <i>legato</i> with mutes, hold through rests	0
9	Bowed, whole tone shake, rests observed	1
K	Bowed, <i>glissando octave</i> , hold through rests	0
S	Bowed —, rests observed	1
N	Bowed √, rests observed	1
J	Snap <i>pizzicato</i>	F
F	Ordinary <i>pizzicato</i>	1
L	Rap on wooden body of instrument with knuckles	F

ancewise might arise if certain playing instructions were juxtaposed, we also permitted an alternative choice of simple *legato* or *pizzicato* to be used during transcription, just in case a playing instruction should turn out to be impossible. In actual fact, this was very seldom needed in transcribing this music for the *Illiac Suite*.

Once this list was prepared, it was possible to associate a numerical value with each instruction, and after so doing, we could use once again

the technique previously employed to establish rhythm and dynamics indications. Thus, it was possible to generate random integers between the values 0, 1, . . . , 15 in order to indicate particular playing instructions for the various instruments. Moreover, as in the cases of rhythm and dynamics, it was also possible to establish how long and to which instruments each playing instruction was to be assigned. Consequently, the same two schemes for horizontal and vertical organization were incorporated into this code as had been used for the rhythm code. In the earlier experiments, a playing instruction was assigned to each instrument independently for any random length up to twelve measures, after which time a new playing instruction was generated for the voice in question. In the revised code, as with the rhythm instructions, simple vertical redundancy was introduced to tie the four voices together a little more closely.

Combination of Rhythm, Dynamics, and Playing Instructions. We have already explained how the representation of rhythm on the Teletype printout was planned. In actual fact, instructions for controlling all three of the above-described musical elements were written as one single computer program so that rhythm, dynamics, and playing instructions were produced successively. The printout for all three was arranged in the most compact form for legibility, with the dynamics and playing instructions for each voice being placed immediately under the rhythmic indication for the same voice. An example of actual computer output produced by the more basic of these two programs for these musical elements is shown in Figure 9,

0101	0101	0101	0101	0101	0101
6 FF=					
0111	0111	0111	0111	0111	0111
S PP(S MF(S MF(L MF(7 MF(L MF(
1010	1010	1010	1010	1010	1010
9 FF)	9 FF)	9 FF)	N F=	N F=	5 F=
1101	1101	1101	0111	0111	0111
J F)	8 F)	8 F)	8 P(8 P(8 P(

Figure 9. Experiment Three: Sample of computer output from the simplest program for rhythm, dynamics, and playing instructions.

while in Figure 10, the transcription of these same results into score form is shown.

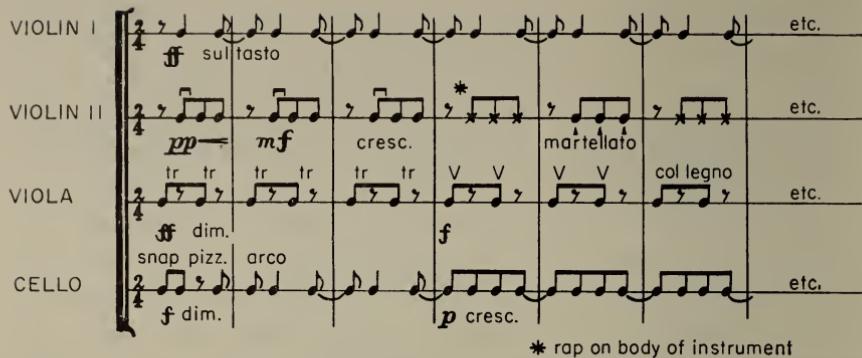


Figure 10. Experiment Three: Transcription of the computer output shown in Figure 9.

Random Chromatic Music. The music shown in Figure 10 lacks any indication of pitch; thus, the next logical step was to develop a program for note selection which would differ from what was done previously with strict counterpoint. For reasons already discussed, we decided to set up a freer basis for note selection and to establish a minimum technique for music writing which would be closer to contemporary practices than strict counterpoint. The first step was to renumber the notes of the scale to include all chromatic steps, black as well as white notes in terms of the piano keyboard. Therefore, for a basic chromatic scale, we used the integers 0-30, which were considered to correspond to a range of two and one-half octaves from C to F#, C being interpreted as the lowest playable C for each of the four instruments. Starting the range of notes with C was convenient, since both the viola and cello have this as their lowest note. The actual process of generating random music involving these thirty-one notes was, of course, extremely simple and involved no more than random-integer generation with the multiplier 31, utilizing the process already described in Chapter 4. At the same time, an improved printout program was written to print the selected notes in letter rather than number notation. In adapting the conventional Teletype machine to this purpose, the letters A through G were used to indicate the notes themselves, the letter X to indicate sharpened notes (F sharp being printed as FX, for example), and primes to indicate the first and second octaves above the lowest octave. For the sake of simplicity, notation for flattened notes was eliminated. The printing of the results was matched to the rhythm code described above, so that the notes and rhythms lined up in a simple four-to-one correspondence. Thus, one line of notes became the equivalent of

three measures of rhythm. Actually, the two basic programs, the one for rhythm and allied musical elements and the other for note selection, were kept separate. This was done simply for convenience, so that each could be revised and made more complex independently and with a minimum amount of rewriting.

Simple Chromatic Music. It was of interest to carry out at least one experiment with random chromatic music to illustrate how a degree of order might be imposed upon this elementary material by simple means. Therefore, in order to write a second type of chromatic music, four compositional rules were imposed on the selection process. The particular rules selected for this purpose were employed because it was felt that these rules would impose a relatively high degree of order with a minimum of actual instructions. The following rules were coded for the Illiac.

Rule 1. This was the melodic skip-stepwise rule already familiar in the strict counterpoint code. This rule was used substantially unchanged except for adjustments required in going from white-note to chromatic music. In the present application, we permitted only melodic movements of a whole tone or less—i.e., a whole tone or a half tone in either direction or a repeated note—to follow a melodic motion greater than a whole tone. On the other hand, any melodic movement obeying the rules in general was permitted to follow a stepwise melodic movement of a whole tone or less.

Rule 2. This was a melodic octave-range rule, also adapted from strict counterpoint. The octave range was set to apply as far back as twenty-four notes, i.e., six measures, maximum. After every three measures, corresponding to one line of printout, had been completed, however, the backward extent of scanning for the octave range was moved up three measures, thus permitting the octave range to be changed in accord with the contents of the last three measures only. This procedure permitted the formation of a new octave range whenever these three last measures contained a melodic substance spanning less than an octave. This octave-range rule, moreover, was affected by the tritone-resolution process to be discussed immediately below.

Rule 3. A rather complex way of resolving tritones was set up which provided the only harmonic control over the musical material being generated. The rule is a simple example of how a reasonable compositional rule can be devised for computer use which helps set up an appropriate control process in a special situation.

Tritones can occur between the four voices in the following ways:

1. A single tritone (only two voices involved)
2. Two independent tritones (all four voices involved)
3. A single tritone with one repeated note (three voices involved)
4. A single tritone with a single note repeated twice (all four voices involved)
5. A single tritone with both notes repeated once (all four voices involved)

(Notes differing by a multiple of an octave were considered equivalent for the purpose of this tritone-resolution rule.)

In the case of the tritones involving no repeated notes, specifically, Cases 1 and 2, the resolution was required to take place by contrary half-tone steps either inwards or outwards, depending upon the result of a binary random choice. For Case 3, the resolution between the single note and one of the two repeated notes was required to be the same as the above, but the other repeated note was permitted to resolve contrary to the first repeated note by a stepwise movement of zero to four half-tones chosen at random. Case 4 was resolved the same way as Case 3, except the fourth voice was permitted to move freely. Finally, Case 5 was resolved by requiring two stepwise motions and two motions of zero to four half-tones chosen at random but in contrary motion to the stepwise movements.

Rule 4. As a last rule, whenever the octave-range rule was violated by a tritone resolution, the resolution was permitted, and the reference point for the octave range was moved up timewise in the musical structure so that the rule would again be satisfied; that is to say, a new octave range was set by the tritone-resolution note. It should be noted that this fourth rule is of considerable interest because it is a simple example of a built-in rule-revision process. The applicable octave range was generated during the course of the first three measures of a given melodic line. It then restricted the range of the melody in accord with Rule 2 until a tritone resolution occurred to force the computer to erase the octave-range limitation in effect and set up a new octave range which had as one of its limits the note farthest back in the melody which was an octave or less separated from the note of the tritone resolution which brought about the revision of the octave range. The operation of this process is shown in Figure 11.

In the computer, the tritone type could be determined after each chord was generated by counting how many times the four voices entered into tritones and then summing. This summing process is illustrated in Table 13. Only in Cases 2 and 3 was there any ambiguity resulting from this



Figure 11. Experiment Three: Illustration of how the octave-range rule is revised by tritone resolutions.

summation process, but this could be resolved by noting that one of the counters was zero in Case 3.

The actual computing process for the application of these compositional rules is shown in condensed form as the block diagram in Figure 12. It is seen that the same general logical outline was required as that used for the strict counterpoint routine previously discussed, and that the initial step in planning the actual program for the Illiac was to decide upon an efficient way of organizing the rules. After the initial entry, which, as in earlier codes, involved the various routine operations required to get the calculations started, the computation cycle was immediately entered at the point at which tritone resolutions were carried out and the test for skip-stepwise motion was made. Naturally, for the first chord, these rules were bypassed, and a random note immediately generated for the first voice and later, upon recycling, for the other three voices as well.

In general, the tritone-resolution operation was divided into three basic operations. Further along in the process, it is observed that the new note in each voice was tested for whether it had occurred in a vertical tritone

Table 13
Experiment Three
Detection of Tritone Combinations by Counting the Number
of Voices in Tritone Combination

Tritone type		V_1	V_2	V_3	V_4	ΣV_i
Case 1	1100	1	1	0	0	2
Case 2	1111	1	1	1	1	4
Case 3	2110	2	1	1	0	4
Case 4	3111	3	1	1	1	6
Case 5	2222	2	2	2	2	8

relationship with any other voices already generated. If not, then the recycling operation was carried out, unless it was the fourth voice being produced, in which case the next part of the program was entered.

On the other hand, if a tritone was found, then a record was made of the voices between which the interval occurred. This was done by adding unit increments into assigned storage locations in the computer. Four spaces in the computer memory were reserved for Voices 1 to 4, respectively, as counting locations for tritone intervals. Two counters were stepped up by one unit each by this operation, as shown in the block entitled "Step two counters" in Figure 12. An example of how this works is shown in Table 14.

After the counting of all the tritones was completed, the second part of the tritone-resolution process was carried out. This was the summation operation already shown in Table 13. The counters in the four locations were simply summed. If the sum ΣV_i was found to be zero, then we knew

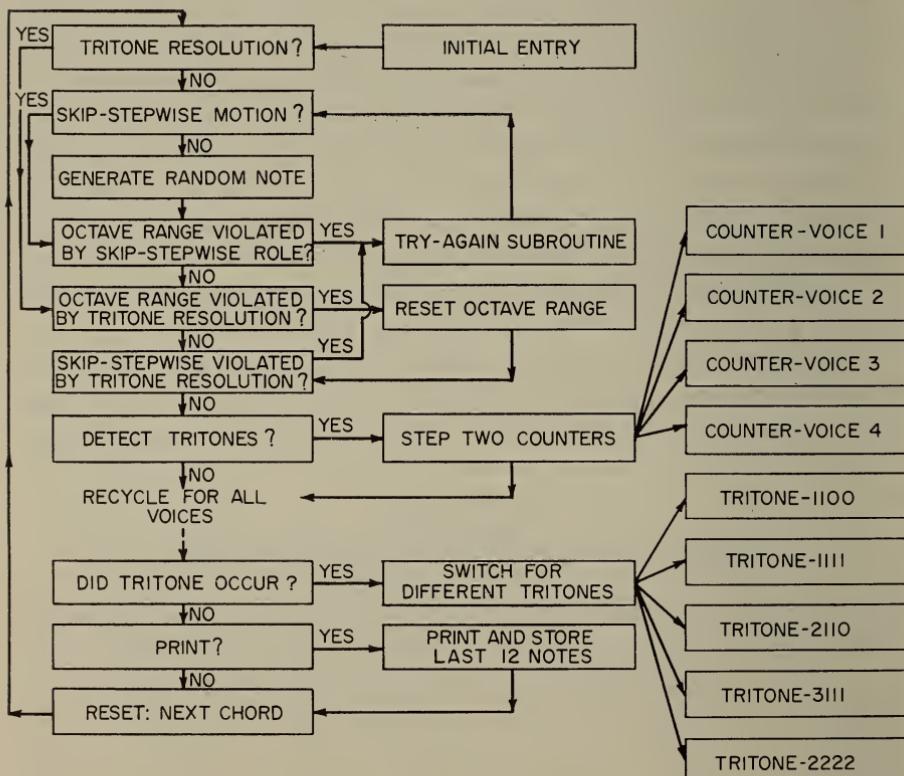


Figure 12. Experiment Three: Block diagram for chromatic writing.

Table 14
Experiment Three
An Example of the Operation of the Tritone Counter

	Storage locations			
	Voice 1	Voice 2	Voice 3	Voice 4
Initially	0	0	0	0
Tritone detected between				
Voices 1 and 3, add	<u>+1</u>	0	<u>+1</u>	0
Result in counter	<u>+1</u>	0	<u>+1</u>	0
Tritone detected between				
Voices 1 and 4, add	<u>+1</u>	0	0	<u>+1</u>
Result	<u>+2</u>	0	<u>+1</u>	<u>+1</u>

that no tritone had occurred. A bypass of the actual tritone-resolution process was then inserted by placing an unconditional transfer order to the skip-stepwise motion test in front of this resolution operation. On the other hand, if this sum was found to be unequal to zero, an unconditional transfer direct to the tritone-resolution routine was inserted immediately before the routine. The particular tritone case was then detected by determining the magnitude of the sum ΣV_i . This was done by a simple additive technique of the type already discussed in the descriptions of Experiments One and Two. In addition, it should be noted that the counters gave a record also of which particular voices were involved. This was essential information, since a simple tritone of the form 1100, for example, can exist in a number of permutations, such as 1100, 1010, 1001, 0110, 0101, and 0011, considering the four digits to represent Voices 1 to 4, respectively, in that order.

After detecting the tritone type by means of the sum ΣV_i , and after a small extra test in case this sum was found equal to 4, a switch was employed to direct the operations to the particular type of tritone resolution required in accord with the rules given earlier. Cases 1 and 2 could be treated by the same basic routine, Case 2 simply involving a repeat of the operation, the operation itself being a simple random binary choice to determine whether the resolution would be inwards or outwards, followed by the actual recording of the required new notes. Cases 3 and 4 could also be treated essentially as a pair, since in Case 4, the fourth voice was allowed to move freely. In these cases, a random choice was first made of which of the doubled voices would be required to move stepwise. Then,

a second random choice, this time a binary choice, was made to determine the direction of the resolution. In Case 3, the remaining voice was then required to move by contrary motion in accord with the rule described above, while in Case 4, a further choice had to be made, namely, to decide which of the remaining doubled voices would be required to move stepwise and which would be permitted to move freely. Case 5, lastly, was a bit more complex. Two random binary choices were first generated in order to determine which voice in each of the two doubled pairs involved in the tritone interval would be selected for stepwise resolution. Then, another binary random choice was generated to determine whether the resolution would occur inwards or outwards. Finally, the remaining voices were required to move in contrary motion to their doubled partners and, hence, in respect to one another. In all five cases, the contrary-motion process made use of a simple five-choice random-integer-choice operation followed by the selection of the correct sign for contrary motion.

Between the test for tritones and the actual carrying out of the tritone resolutions if tritones were found, it was necessary to insert two other operations. As shown in Figure 12, the first of these was a set of printing instructions which was activated by a counter after every twelve chords had been formed. This operation caused the computer to punch tape which would result in one line of printed output. Simultaneously, this completed line of output was stored in the computer memory, overwriting the previous line of output, so that the octave-range rule could be applied backwards for three measures beyond the new line to be compared. In addition, independently of these printing instructions, counters and various other routines were set so that the next chord would be generated.

The operation of the remaining sets of instructions is rather obvious in the light of the detailed descriptions given of similar rules applied in Experiments One and Two. The skip-stepwise-motion routine was set up entirely similarly to the related strict counterpoint rule, with the necessary modifications required to permit chromatic motion. The octave-range rule likewise was adapted from the operations used for the strict counterpoint program. The resetting of the octave-range by the tritone-resolution process was extremely simple and involved merely the resetting of two storage indices used for this computation. Lastly, the try-again subroutine was the same as that used before, except for minor bookkeeping details.

Interval and Tone Rows. One simple way to initiate a study of the relationship between entropy and melody is to consider a twelve-note melody. If we were to tabulate all possible twelve-note melodies, we would see that

the number of possible melodies could be expressed in terms of elementary permutation theory. Thus, if we require only that the twelve notes be selected from twelve possible tones with no restrictions in regard to repeats of tones being required or forbidden, we observe that we have defined the condition of random music within this limited situation. If this condition applies to all twelve notes, 12^{12} melodies are theoretically possible—an enormously large number. This is a situation of maximum entropy, or information, content in terms of the choice process, but it is not necessarily the condition of maximum entropy content in terms of tonality. This can be arrived at by a somewhat different procedure. Let us first note that among all types of melodies in this random situation, the variety of melodic types extends from the case in which all twelve tones of the melodies are required to be the same to the case in which all twelve tones are required to be different. At one end of this spectrum, where all the tones are required to be the same, there exists a minimum of entropy—i.e., a maximum of redundancy—because the total possible number of melodies is just twelve if we permit full freedom of choice for the first tone and just one if we restrict this choice to one specific pitch. This is obviously totally established tonality—twelve repeats of the same tone. At the other end of the spectrum, however, where the twelve tones are all required to be different, each tone of the chromatic scale is sounded just once in some specific order which may or may not be randomly produced. This is, of course, the tone-row concept first significantly exploited by Arnold Schönberg. The fact that the specification of a tone row is arrived at by precisely the opposite technique as that of “ultimate tonality,” namely, by requiring that the tones be different as contrasted to requiring them to be the same, suggests the thought that a tone row is not just simply an atonal device, but rather that it is a specific “antitonal” construction which, within its own length, is deliberately devised to exclude the type of redundancy required to set up tonality. Atonality, as a term, is therefore better applied to the situation of random music rather than to constructions such as tone rows. In this way, tone rows present rather interesting musical properties, since they represent not only a highly restrictive and easily managed technique for reducing the entropy content of random music, but also an extreme condition in terms of elementary permutation theory and a computational extreme for counteracting conventional tonality. It can be readily shown that the total number of possible tone rows is $12!^4$, which, although it is a relatively large number, is considerably less

⁴ $12! = 12 \cdot 11 \cdot 10 \cdot 9 \cdot 8 \cdot 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 479,001,600$.

than the total possible number of twelve-note melodies. Moreover, in everyday practice, the total possible number of such melodies is effectively much less than this, since one type of redundancy commonly employed in tone-row compositions is that based upon permutation techniques such as inversion, retrogression, and the like. It is important to note also that transposition, of course, does not alter the basic character of tone rows any more than it does other types of melodic lines. This is equivalent to saying that there are really only $11!$ rather than $12!$ possible tone rows, since the particular pitch level at which the row is played is of secondary consequence compared to the actual profile of the row. In view of what we said earlier in Chapter 2, we may then infer that the melodic profile of a tone row depends *not upon a succession of the twelve tones of the chromatic scale* as such, as is often commonly supposed, but upon the *mutual interrelationships between the twelve notes in terms of intervals*. Since in theory at least all the twelve tones are presumed to be of equivalent importance, the melodic profile is, therefore, a series of intervals between successive notes in which any one or all of the twelve notes can be selected as a reference point. Once a reference point is defined, it is observed that the tone row consists actually of a series of all the possible melodic intervals considered not in succession but in relation to this reference tone. To make this point clear, we may arbitrarily choose the first note of a tone row as the reference tone and then illustrate by an example. In Figure 13, a typical tone row is

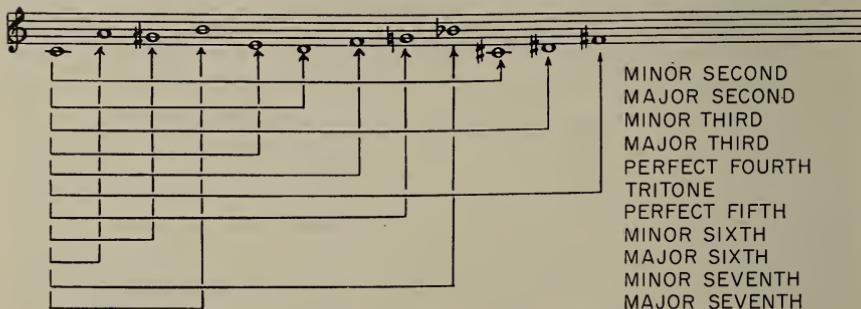


Figure 13. Experiment Three: Illustration of how all the possible intervals occur in a row of twelve different tones.

shown. It contains all the possible melodic intervals from a minor second to a major seventh *in relation to the first note of the row, C* (or, for that matter, in relation to any other note of the row). Moreover, each one of the intervals, which we may define now as the difference $N_i - N_1$, using the

notation adopted previously for notes and intervals, occurs *once and only once*, but the intervals found between successive tones $N_i - N_{i-1}$, $N_{i+1} - N_i$, etc., may occur more than once. Thus, repeats of *successive* melodic intervals are not forbidden.

On the other hand, we can also write rows in which repeats of successive melodic intervals of the type $N_i - N_{i-1}$ are forbidden, but repeats of reference intervals of the type $N_i - N_1$ are permitted. This produces what we can call an *interval row* as opposed to a tone row. The interval row perhaps is a new type of melodic structure, which should present some interesting musical properties, particularly in light of the experiments to be described under Experiment Four, in which the properties of different varieties of melodic intervals, both successive and long range, are examined in greater detail.^{4a} A typical interval row is shown in Figure 14. It should

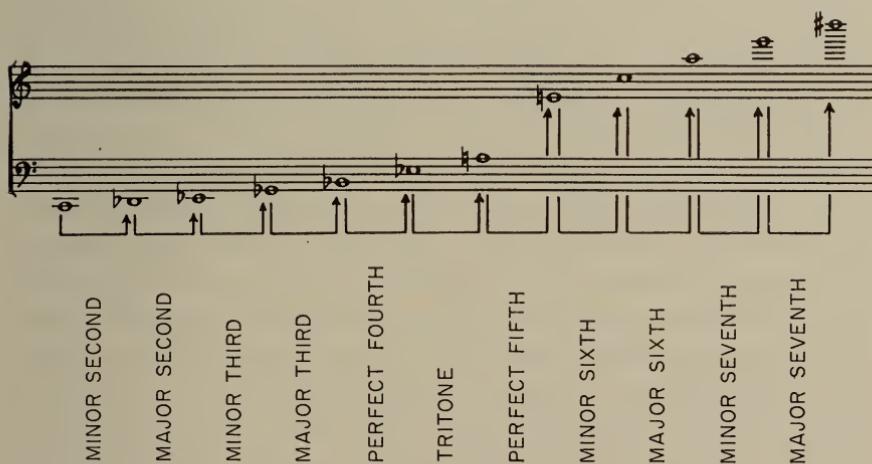


Figure 14. Experiment Three: A typical interval row, not showing transpositions to reduce span of row.

be noted that actual tones or their octave equivalents can be repeated in an interval row in contrast to the tone row. It is not necessarily a situation of greater redundancy than that of tone rows, however, since both permit the same total number of possible choices. Redundancy in terms of tonality is greater in the case of interval rows, but whether this is, in terms of musical meaning, a situation of greater redundancy depends on whether successive or referential intervals are to be considered the more significant in the con-

^{4a} It might be noted parenthetically that since working on this computer music, one of the present authors (L. A. H.) has written an extended piano composition based in part on this notion of an interval row.

struction of melodies. In any event, we may note that both tone rows and interval rows have similar specific constraints imposed upon them in regard to repetition of choices. In both choice processes, eleven choices of intervals are permitted initially, but each time an interval is chosen, it is excluded from the remaining choices to be made. This is analogous to the problem of choosing colored balls from a box and *not replacing them after each choice*, as opposed to a random succession which occurs when the balls are replaced. It is by analysis of this process that the value of $11!$ is arrived at. Within this particular choice process, we may also observe that the difference between a tone row and an interval row is the following: A tone row is an *i*th-order Markoff process, while an interval row is a zeroth-order Markoff process in terms of how the intervals are evaluated. This difference is made clear if the reference states in Figures 13 and 14 are compared.

Once we agree to the foregoing analysis of the properties of interval rows and tone rows, it becomes simple to write a program for the computer for generating these types of musical structures, of which the tone row, at least, is at the present time a frequently used compositional device. The concluding part of Experiment Three, therefore, consisted of three related programs for the Illiac for generating (1) interval rows of the type described above, (2) tone rows, also of the type described above, and (3) tone rows of a somewhat more restricted character. These last tone rows were written at the suggestion of Robert Kelly, of the School of Music at the University of Illinois, and are rather freely based upon a pedagogical approach to music writing developed by this author for teaching counterpoint and harmony from a contemporary viewpoint in beginning composition courses.⁵ The method combines certain properties of tone rows with adaptations of a number of the rules of strict counterpoint. The rules employed by Kelly are given in the cited reference. Our adjustments of these rules for computer processing are the following:

Rule 1. The theme should consist of thirteen notes using all twelve tones, with the first and last notes being C.

Rule 2. A span of a major tenth is set as the limit for the melodic range.

Rule 3. Progressions of major sevenths, minor sevenths, and tritones are forbidden.

Rule 4. No tied-over notes are permitted. This rule was automatically taken care of the way the tone-row generation scheme was set up (see below).

⁵ R. Kelly, *Theme and Variations, A Study of Linear Twelve-tone Composition*, Wm. C. Brown Co., Dubuque, Iowa, 1958, pp. 2-4.

Rule 5. Chromatic progressions in the same direction involving three or more consecutive notes are forbidden.

Rule 6. The skip-stepwise rule is modified to permit two skips in the same direction.

Rule 7. Consecutive sets of three notes that constitute a triad may occur at random.

Rule 8. No tritone is permitted for $N_i - N_{i-1}$ unless $N_{i-1} - N_{i-2} = \pm 1$, or, alternatively, 5 or 7 in the same direction as the tritone.

Rule 9. The use of sequences is prohibited. In other words, $N_i - N_{i-1}$ cannot equal $N_{i-2} - N_{i-3}$, and so on.

It can be seen immediately that a block diagram for a testing procedure similar to those shown already for Experiments Two and Three could be designed, and from this a testing program written which would include the usual type of try-again subroutine. This testing procedure was written as a subroutine which was grafted onto the basic interval and tone-row-generation program described below. Since this subroutine involved no basically different programming techniques from those already described, it seems unnecessary to consider its details, since these largely duplicate what we have already discussed.

The Generation of Interval and Tone Rows. The production of interval and tone rows was quite simple to carry out in the Illiac. The main routine for this process was designed as follows: To make the note-selection process efficient, a small table containing numerical representations from one to twelve was stored in the computer. These were considered to be numerical representations for all the possible intervals from a minor second to an octave which simultaneously gave a count for each interval of the number of half-tone steps within the interval. The unison or repeat (numerical representation = 0) was omitted from this table, since it was decided to apply Kelly's procedure of writing *cantus firmi* of thirteen notes, of which the first and, in the tone rows, the last would be set automatically as C. An interval was then selected at random from the above-described interval table and added to the interval or tone-row representation being formed to obtain the first tone in the sequence of the interval or tone row. This interval was then removed from the table, and all entries below it were moved up one line and the process repeated. Thus, we started with twelve possible choices, then went successively to eleven choices, ten choices, and so on, until we were left with only one choice for the final interval. When an interval or tone row was completed, it was then printed out and the computer reset to perform the operation all over again.

The way in which the intervals being selected at random were added to the row being formed determined whether or not an interval or tone row was formed. In view of our earlier discussion, the difference is seen to be one of minor coding changes. To produce an interval row, the numerical representation of the interval just chosen was added to the previous note in the row to produce a new note in the row separated from the previous note by the interval in question. However, since we always *added* new intervals to a row to give upward progressions, it became necessary to employ also an octave-subtraction process, because otherwise the span of the row would soon exceed the set limit of two and one-half octaves of the chromatic scale. This is in accord with normal practice, however, for subtracting an octave from a note does not disturb its primary functional relationship to a melodic reference point such as the first note of the melodic line. What we actually did in the generation of interval rows was to make a binary choice which determined whether we would simply *add* the interval to the previous note and, therefore, move upwards in pitch or, alternatively, *add* the note and then *subtract* one octave to produce a note lower than the last. Secondly, it was also necessary to test for whether the limits of the chromatic scale being used were being exceeded in spite of this. Therefore, a test was also made to determine whether this was so, and if necessary, and adjustment of the movement in the proper direction, either upwards or downwards, as the case might be, was then made. Finally, it was observed that an interval row starting on C always of necessity ends on F sharp.

To produce tone rows rather than interval rows, the interval randomly selected from the table was simply added to the *first note* C of the row, rather than to the previous note in the row. This automatically produced a tone row rather than an interval row. Otherwise, the whole process was kept exactly the same, except that the octave was removed from the table of intervals, since the choice of this interval would lead to the repeat of a *tone* (or its octave equivalent) within a tone row. This would be a violation of the requirements for such a row. In generating simple tone rows of thirteen notes according to the chosen plan but with no restrictions, we simply arbitrarily made this the *last* interval to be selected, so that the row this time would of necessity have C or its octave equivalent as its final note. In the case of the restricted tone rows, in addition to writing a subroutine which would apply the rules given above, we could simply eliminate the octave from consideration as a possible interval.

Tone-row Permutations. A basic procedure in writing tone-row music is

to perform geometrical permutations upon the row being utilized to compose a piece of music. There are many possible geometrical operations upon tone rows which can be used, but of these the three simplest and most frequently used are inversion, i.e., writing each interval downwards instead of upwards, retrogression, i.e., writing the row backwards, and lastly, inversion of the retrograde form of the row.^{6, 7} In the computer, these geometrical permutations were most easily carried out as part of the printout procedure. In this way, they are handled rather analogously to the process of transposition which we shall consider under Experiment Four. We arranged the printout to contain in a block of four lines: (1) the row itself, (2) its inversion, (3) the retrograde form, and (4) the retrograde inversion, each printed one under the other. The printing of these four forms of a row was just a matter of format arrangement for the printout routine. The row itself was first printed a symbol at a time, each numerical representation of a note being converted into its letter equivalent. Secondly, the inversion of the row was formed by changing the signs of the successive intervals in the row and then printing out the letter equivalents of the notes of the inverted row. Lastly, the two retrograde forms were produced by printing out these two forms of the row again, but selecting the notes in reverse order.

Experiment Four

The objectives of Experiment Four differed from those of previous experiments, since it was our purpose in this new experiment to generate samples of music based upon certain more abstract formulations which we believed might eventually be related to structural devices in musical composition such as tonality and melodic profile. In the first three experiments,

⁶ Strictly speaking, one should differentiate between retrograde inversions and inverted retrograde rows, since these two variations, although they have the same profile, may not occur at the same pitch, depending upon the last note in a row. Thus, a tone row beginning on C and ending on E, for example, will have a retrograde inversion beginning on A flat and ending on C, while the inverted retrograde form of the row will start on E and end on G sharp. The recognition of this difference, of course, produces a modulatory technique in tone-row writing. In our samples, however, which begin and end on C, this problem did not arise because of symmetry considerations. This also happens to be true for the interval rows as well, since these must of necessity end on F sharp, which is likewise a situation of symmetry.

⁷ It is perhaps also desirable to note, in passing, that these basic techniques are, of course, by no means unique to row writing. As is well known to musicians, all these, as well as many other geometrical operations, are common procedures in many types of well-established musical forms such as, for example, in canons and fugues.

and particularly the first two, we were primarily concerned with the generation of music recognizable in many respects as falling within the traditional framework of compositional procedures. In Experiment Four, this was no longer intended to be the case. In this last set of experiments, simpler and perhaps more fundamental means of musical construction were investigated than those studied previously. This was done in an attempt to find more inclusive concepts to work with, and, in particular, concepts which might be thought of as geometrical analogs of musical form.

The fundamental geometrical picture selected was an abstraction of the calculating technique used in the three previous experiments. The generation of four-part musical structures was pictured as a restricted random-flight problem in which four trajectories are traced simultaneously upon the rather unusual coordinate system of pitch versus time. This geometrical structure, illustrated in Figure 15, can be subjected to mathematical defini-

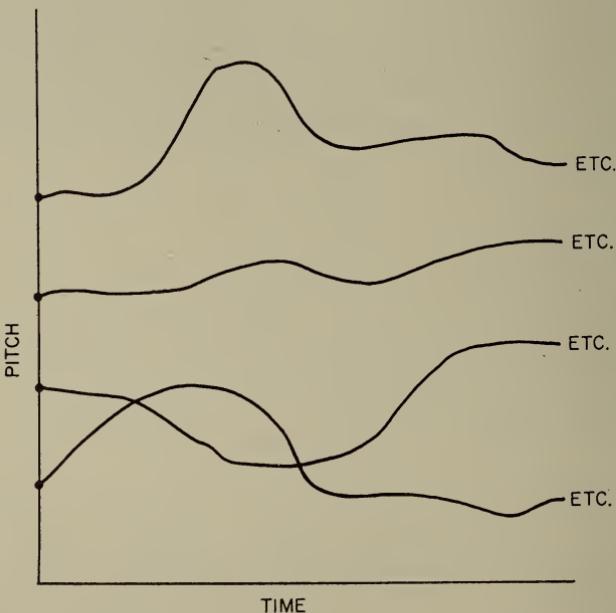


Figure 15. Experiment Four: A four-voiced musical texture pictured as a random-flight problem.

tion, as we have already indicated in Chapters 2, 3, and 4. It is particularly characterized in terms of Markoff chain processes, i.e., sequences of events in which the choice of each new event can be made dependent upon pre-

vious events; or, in musical terms, the choice of each new note or interval in a given melodic line can be made dependent upon previous notes or intervals in the same melodic line. Utilizing this simple picture, we wrote computer programs for generating a series of samples of what we may call *Markoff chain music*.

Harmonic and Proximity Functions in Melodic Construction. If we recall the dependences of melodic construction and tonality upon both the successive and long-range intervallic relationships discussed in Chapter 2, we may now note that successive melodic intervals bear three significant characteristics which *may* be used to impart formal structure to a melody: (1) Melodic intervals can be related to the overtone series so that we recognize a harmonic function in melodic construction. Thus, the outlining of a triad, C-E-G for example, imparts a sense of C major, or, at least, of the C-major chord, and a sense of relatively high tonal order. On the other hand, a sequence such as F-A flat-D is more ambiguous and conveys a number of possible implications in terms of structural function, other factors being equal. (2) Melodic intervals may also be characterized by their absolute size. Thus, stepwise melodic progressions such as half-tone or whole-tone intervals seem to impart more order than larger skips such as sevenths, octaves, and larger intervals. (3) We may note that many well-constructed melodies consist of sequences of intervals which balance the tendencies to order and disorder by balancing harmonic clarity with ambiguity, and stepwise melodic movements with larger skips. Even in music in which these principles seem to operate minimally, as in certain types of contemporary music, the presence of both functions, which we may now call *harmonic* and *proximity* functions, can be traced. In a way, this can be said to be true even indirectly of the steps resorted to by composers in negating tonality by devices such as twelve-tone rows.

Relation to Musical Structure. These two functions can also be examined somewhat more broadly than just in terms of relatively short-range melodic relationships. In terms of more recent musical styles, these two functions of harmonic and intervallic structure also form the basis of many of the more searching analyses of musical forms such as, to cite two representative and well-known authors, those by Hindemith⁸ and by Schenker,⁹ whose

⁸ P. Hindemith, *Craft of Musical Composition—I. Theory*, rev. ed. (trans. by A. Mendel), Associated Music Publishers, Inc., New York, 1945.

⁹ H. Schenker, *Neue Musikalische Theorien und Phantasien*: vol. I, *Harmonielehre*, 1906, vol. II, *Contrapunkt*, 1910 and 1920, vol. III, *Die Freie Satz*, 1935, Universal Edition, Vienna.

concepts have been presented in English by Katz,¹⁰ Salzer,¹¹ and Jonas and Borgese.¹²

Both Hindemith and Schenker distinguish the harmonic and melodic element, and both acknowledge the dependence of the harmonic elements upon recognizable relationships to the harmonic series and of the melodic element upon stepwise motions. Thus, Hindemith stresses the importance of "root-progressions," of which the strongest harmonically are shifts, such as fifths, fourths, and thirds, in giving convincing harmonic movement to musical passages.¹³ He also remarks that a smooth and convincing melodic outline is achieved only when these important points form progressions in seconds.¹⁴

However, Hindemith confines himself almost wholly to successive musical relationships, leaving the question of larger structures relatively unexplored; therefore, we might also consider Schenker's ideas, which are somewhat more ambitious in scope. He suggested that elements of a musical structure could be compared to a series of structural layers, starting from a "foreground" which is the complete work as we hear it to a "background" structure in which only a skeletal framework of the music is left. In analyzing a piece of music, Schenker proposed that these layers can be successively stripped away to reveal the basic framework, which he called the *ursatz* and which he thought consisted of two basic elements. The first we can interpret as the essential harmonic root movement of a whole structural unit, such as a song or sonata-allegro form. He postulated that all satisfactory musical forms, even lengthy ones, could be simplified to one of several simple root progressions, such as I-III-V-I and I-IV-V-I, and was able to build quite a good argument for this idea by utilizing exclusively examples from the works of the master composers of the eighteenth and nineteenth centuries, the only repertory he considered admissible. The second element of the *ursatz* proposed by Schenker is the *urlinie*. This is a significant essential descending melodic line contained in every "acceptable" piece of music. The *urlinie* is always basically simple, being a sequence such as, in C, G-F-E-D-C. Complex musical structures are then built up by elaboration upon these two elements of the simple skeletal

¹⁰ A. Katz, *Challenge to Musical Tradition*, Alfred A. Knopf, Inc., New York, 1945.

¹¹ F. Salzer, *Structural Hearing; Tonal Coherence in Music*, Albert and Charles Boni, Inc., New York, 1952.

¹² H. Schenker, *Harmony*, from *Harmonielehre*, ref. 9 above (ed. by O. Jonas and trans. by E. M. Borgese), University of Chicago Press, Chicago, 1954.

¹³ Hindemith, *op. cit.*, chap. IV in particular.

¹⁴ *Ibid.*, p. 193.

framework. The principles of the theory are supposed to be in operation within sections of a piece of music in a subsidiary role as well as over the whole musical structure. Many examples of how the Schenker system of analysis can be applied are given in Salzer's book; it is to be noted that Salzer goes to considerable effort to circumvent Schenker's dogmatic attitudes regarding the absolute superiority of Germanic music from 1700 to 1900, which has marred the impression of his work in the eyes of some critics.¹⁵ It is not our purpose to become involved in detailed criticisms of methods of musical analysis, but only to extract from them several ideas which were useful in planning the experiments to be described and which would be of paramount importance in any future experiments in which more elaborate musical structure might be produced. The first is the concept of structural frameworks, both harmonic and melodic. This concept is useful because it provides a differentiation between structural points of greater or lesser significance. A working technique for structural assembly is thus provided that permits the assigning of a greater or lesser value to significant musical elements, particularly in relation to the time scale of the musical structure. It is observed that this factor is almost entirely missing in the computer music produced in the first three experiments.

The second basic point is the emphasis on a distinct harmonic element and a distinct melodic element in the basic structure. This does seem to be in general accord with our definitions of a harmonic and a proximity function, and suggests a method of elaboration of the techniques shortly to be described under Experiment Four for the construction of more complex musical structures. The particular emphasis by analysts such as Schenker on the practices of the eighteenth and nineteenth centuries, which established for them a standard for clarity of tonality, need not be mistaken for an absolute necessity. If we consider musical structures antecedent to 1700 as a source of ideas, we can observe that this music was written before the concept of tonality was clearly defined. Thus, in view of the emphasis on contrapuntal rules rather than harmony in this earlier music, the proximity-type function might be found to play a more preponderant role here. Moreover, closeness of intervals seems to have been used as a guiding principle, not only for simple step-by-step techniques, such as the skip-stepwise rule of counterpoint, but also implicitly as an important structural device in,

¹⁵ M. Mann, "Schenker's Contribution to Music Theory," *Music Review*, 10:3-26, 1949; R. Sessions, "Heinrich Schenker's Contribution," *Modern Music*, 12:170-178, 1935; R. Sessions, "Escape by Theory," *Modern Music*, 15:192-197, 1938. This last article in particular is a highly critical review of Schenker's *Die Freie Satz*.

for example, much primitive music, non-Western music, and in older Western art music such as plain chant. Coherence is not infrequently brought about by melismatic weaving about certain structurally important melodic notes, or even around one fixed tonal center. This might also be true in certain types of contemporary music as well. Thus, Hindemith,⁸ for example, permits in his system of composition a structural dependence on proximity, or neighbor-note relationships, as well as on strongly harmonic relationships such as the fifth, fourth, and third.

Table of Weighted Probabilities. In order to institute an analytical study of these processes, we first defined a harmonic function x_j and a proximity function y_j . Then, to investigate how these functions might operate to produce melodic structures, a series of experiments was carried out in which Markoff chain processes were used as a basic computing technique. The first requirement was a reference table of weighted transition probabilities stored in the computer, to be utilized as required. It should be noted that, in essence, this had been done all along in previous experiments, but in these earlier experiments, the transition probabilities were usually either simple random-choice situations, i.e., all choices were weighted equally, or else some of the choices were simply forbidden and thus given zero weight. The counterpoint rules, for example, often amount to no more than just this. The second requirement we imposed was that this table must contain as few bits of basic information as possible, but that this information be of as fundamental significance as possible. In this way, we hoped to investigate how to avoid the problem of high-capacity information storage, which currently presents such a major difficulty in many other data-processing problems such as language translation by machine. Therefore, a table was constructed for arranging all successive melodic intervals from the unison (melodically, the repeat) to the octave. This is shown in Table 15, where the various possible melodic intervals are listed first in descending order of consonance from the repeat to the tritone. For the purpose of machine calculations, we associated with each interval a value of a stochastic variable v_j , which runs from 0 for the unison, or repeat, to 12 for the octave. These are the values we could add or subtract to each *tone* of a melody to obtain the next *tone* of the melody, which resulted as a consequence of moving through the interval represented by the variable. A melody could, therefore, be symbolized by a sequence of values $v_i = v_1, v_2, \dots, v_n$, added successively to the first note of the melody. In the present work, the note C was normally selected as an arbitrary starting point. Once again, as in previous studies such as that of interval and tone rows, intervals,

Table 15
Table of Functions for the Generation of
Markoff Chain Music in Experiment Four

Interval	Stochastic variable v_j	Harmonic function x_j	Proximity function y_j	Combined function $z_j = x_j + y_j$	
Unison	0	13	13	26	
Octave	12	12	1	13	
Fifth	7	11	6	17	
Fourth	5	10	8	18	
Major third	4	9	9	18	
Minor sixth	8	8	5	13	
Minor third	3	7	10	17	
Major sixth	9	6	4	10	
Major second	2	5	11	16	
Minor seventh	10	4	3	7	
Minor second	1	3	12	15	
Major seventh	11	2	2	4	
Tritone	6	1	7	8	
$\sum_{j=0}^{12} x_j = 91$		$\sum_{j=0}^{12} y_j = 91$		$\sum_{j=0}^{12} z_j = 182 = 2 \times 91$	
[$x_j = x(v_j)$]		[$y_j = y(v_j)$]			

rather than tones, were used as the determining functions for assembling melodic structures.

As shown in the second column of Table 15, the harmonic function x_j was given values which run arithmetically from 13 to 1 in descending order of consonance. This simple arithmetic ordering was defined as an *unperturbed* set of weighted probabilities for the harmonic function. If these values are used directly for interval selection, this brings about, *on the average*, the selection of repeats thirteen times as frequently as the tritone, of octaves twelve times as frequently as the tritone, and so on. It was believed that this simple arithmetic scale of relative weights for the harmonic function would be an adequate representation of a neutral or mean position in terms of the imposition of tonal order. It is upon these values that we had to operate in order to perturb the *mean* harmonic texture, and thus achieve a higher or lower average degree of tonality.

In the third column, a similar set of values is tabulated for the proximity function y_j . It is seen that these values run arithmetically from 13 down to 1, as do the values for x_j , but that the *order* of the values relative to the

stochastic variable v_j has been changed, so that the highest unperturbed weight is assigned to the repeat and the next highest weights successively to the minor second, the major second, and so on.

In the last column, values are tabulated for a third function, z_j , which is simply the sum $x_j + y_j$. This combined function was used in certain experiments to show how a simple combination of the two individual functions might be carried out. It is interesting to note that the two functions combined in this way yielded a scale of weighted probabilities that is, perhaps, not too different from conventional melodic writing. A statistical count of melodic intervals in appropriate examples of music might be of interest compared to this column.

Lastly, at the bottom of Table 15, there are given values for the sum of the weights for each function. These values were required for computer calculations to determine the frequency with which each interval occurred on the average. Thus, if x_j only was used to generate music, the relative frequency with which major thirds turned up, for example, was

$$x_4 / \sum_{j=0}^{12} x_j = 9/91 \approx 0.1$$

or approximately ten per cent of the time.

Generation of Integer Weights. The values given in Table 13 were entered into the computer to constitute a set of weighted probabilities stored in the memory. As shown in Table 16, in which the information given in

Table 16
Assignment of Integer Weights to a Stochastic Variable

Stochastic variable, v_j	Integer weight, w_j
0	w_0
1	w_1
2	w_2
.	.
.	.
.	.
j	w_1

$j = 12$ in the calculations of Experiment Four

Table 15 is generalized, we could then assign any desired weights, $w_0, w_1, w_2, \dots, w_{12}$, to the possible intervals from unison to tritone. Having done this, we could then determine the sum of these weights, i.e.,

$$\sum_{j=0}^{12} w_j = W \quad (15)$$

and generate a random integer R within the limits $0 \leq R \leq W - 1$, as described previously under random-integer generation. The value $-R$ was then obtained and the weights w_j were added successively to this quantity *in sequence* until the sum

$$-R + \sum_{j=0}^R w_j$$

became positive. At this point, we took the value of the stochastic variable associated with the last weight added as the desired one. By this scheme, it is evident that the probability for obtaining w_j is w_j/W .

Changing the Weights of Transition Probabilities. The texture of music being generated by means of transition probabilities can be varied by changing the magnitudes of these probabilities in relation to one another. Thus, the "reference," or unperturbed, probabilities shown in Table 15 were operated upon in several different ways in Experiment Four, in order to see how changes in these weights would be reflected in the musical output. In practice, given values such as those shown in Table 15, there were essentially only two direct operations which could be effected, namely, making the weights more equal or, conversely, more unequal. Taking the harmonic function x_j as an example, it was possible to *subtract* unity successively from the unperturbed probabilities to remove the less heavily weighted probabilities one by one, the operation upon each x_j being terminated whenever its value became zero. In this way, we could gradually restrict the successive steps in a melody to the consonant intervals and eventually to just the octave and repeat, at which point fixation upon one note in the scale and, hence, upon a certain tonality was achieved. This technique could, therefore, be used to generate a type of harmonic drive toward a tonal center, while the reverse process is used to move away from a highly organized tonal center toward a less well-defined tonal organization. Naturally, it would also be possible to subtract or add larger increments than unity for more sudden shifts in musical texture.

A shift toward random music, in which all intervals are equally probable, was also easily achieved. This was accomplished by *adding* unity successively to the unperturbed transition probabilities until each one achieved the maximum value, namely, thirteen, at which point the value was fixed. All values for the weights, therefore, tended toward equality. Again, it would be equally possible to reverse the process and make the weights successively less equally weighted.

Both these operations were actually carried out, and the results thus produced were utilized to form the first two sections of Experiment Four.

Moreover, some additional experiments of this type, not included in the *Illiac Suite*, were also completed. It was also possible to subtract integer values from equally weighted transition probabilities—those for random music each having a value of thirteen—in reverse order, thus reducing the value for the melodic repeat first and the tritone last. In this way, we were able to obtain a set of inverse transition probabilities for x_j which produced music in which tritones occurred thirteen times as frequently as melodic repeats. This process was continued until all the intervals except the tritone were eliminated. This last was a species of music as highly restrictive as that which consists only of repeated notes. Samples of this output were generated by means of the Illiac, but were not included in the *Illiac Suite* because the previous results illustrated sufficiently well the general techniques of altering the weights of transition probabilities. The various operations described for altering transition probabilities are shown in Table 17.

Table 17
Changing the Weights of Transition Probabilities,
Utilizing the Harmonic Function, x_j ,
as an Example

	Addition		Addition		Inverse subtraction	
	Subtraction	x_j	subtraction	x_j	Inverse addition	
Repeat	1 2 3 4 ... 11 12	13	13 13 ... 13	13	12 ... 2	1 0 0 ... 0 0
Octave	0 1 2 3 ... 10 11	12	13 13 ... 13	13	13 ... 3	2 1 0 ... 0 0
Fifth	0 0 1 2 ... 9 10	11	12 13 ... 13	13	13 ... 4	3 2 1 ... 0 0
Fourth	0 0 0 1 ... 8 9	10	12 12 ... 13	13	13 ... 5	4 3 2 ... 0 0
Major third	0 0 0 0 ... 7 8	9	10 11 ... 13	13	13 ... 6	5 4 3 ... 0 0
Minor sixth	0 0 0 0 ... 6 7	8	9 10 ... 13	13	13 ... 7	6 5 4 ... 0 0
Minor third	0 0 0 0 ... 5 6	7	8 9 ... 13	13	13 ... 8	7 6 5 ... 0 0
Major sixth	0 0 0 0 ... 4 5	6	7 8 ... 13	13	13 ... 9	8 7 6 ... 0 0
Major second	0 0 0 0 ... 3 4	5	6 7 ... 13	13	13 ... 10	9 8 7 ... 0 0
Minor seventh	0 0 0 0 ... 2 3	4	5 6 ... 13	13	13 ... 11	10 9 8 ... 0 0
Minor second	0 0 0 0 ... 1 2	3	4 5 ... 13	13	13 ... 12	11 10 9 ... 0 0
Major seventh	0 0 0 0 ... 0 1	2	3 4 ... 13	13	13 ... 13	12 11 10 ... 1 0
Tritone	0 0 0 0 ... 0 0	1	2 3 ... 12	13	13 ... 13	13 12 11 ... 2 1

↑ Fixed tonality ↑ Unperturbed distribution ↑ Random music ↑ Inverse unperturbed distribution ↑ Tritone music

Obviously, the same types of operations can also be carried out on the proximity function y_j , or for that matter on any other arbitrary selection of transition probabilities.

Zeroth-order Markoff Chain Music. In a monodic pattern, if successive notes are isolated and considered as such, it is possible to write music in which the choice of N_n is entirely independent of its relationship to note N_{n-1} . As we have seen, the mathematician calls a structure of this general type a zeroth-order Markoff chain—in particular species of random flight in which each successive event is independent of its relationship to the immediately preceding event. This would be zeroth-order Markoff chain music with respect to notes. A primitive type of relationship between notes, however, will be defined if we write zeroth-order Markoff chain music with respect to successive melodic *intervals* rather than *notes*, i.e., if we make I_n independent of I_{n-1} . Since we have already indicated in Chapter 2 that we prefer to think of melodies in terms of successive intervals rather than notes, let us define this as zeroth-order melodic writing. In zeroth-order melodic writing, then, a priori transition probabilities, such as those given in Table 13, are defined to permit moves from one note to the next and to permit the generation of sequences of melodic intervals. As we have already noted, the values assigned these transition probabilities establish the character of the music, but except for the unique case already considered of assigning a transition probability of unity to the unison (i.e., the repeat melodically)—or its harmonic equivalent, the octave—and zero to all other intervals to give rise to patterns of repeated notes, such music is intrinsically atonal, since there is nothing in the nature of zeroth-order transition probabilities to establish specific tonal centers. In zeroth-order Markoff chain music, a melody starting on C, for example, can move anywhere conditioned only by transition probabilities, and if such a melody arrives, for example, on F sharp, there is no compulsion to return to C or to any other note. Moreover, even if we weight most heavily the probabilities for consonant leaps, like rising and falling fifths, all that happens is that the apparent harmonic movement becomes slower. Conversely, the more we weight the more dissonant intervals, the more rapidly we tend to experience oscillations between more distantly related tonal centers. Thus, tonality as such does not exist in such a texture except as the result of chance. Zeroth-order Markoff chain music is the simplest type of order placed upon random-note music, but it still contains within its species all types of music from tonal through atonal and finally to antitonal music, such as certain types of tone-row composition. All such more highly ordered systems therefore arise solely as the result of chance events and not as a result of design. We can also conclude that the consonance or dissonance of successive melodic intervals depends on the average upon how

transition probabilities are weighted. Zeroth-order Markoff chain music can appear highly consonant or highly dissonant or any mixture in between. Thus, the simplest order is imposed upon random-note music by weighting certain zeroth-order transition probabilities more heavily than others. In terms of information theory, random-note music is characterized by maximum entropy content, while the weighting of transition probabilities in any direction, as in Table 17, for example, decreases entropy by increasing redundancy. However, it should also be noted that the degree of consonance or dissonance has nothing to do with tonality as such, since the essential characteristic of tonality, in view of our comments in Chapter 2, is the recall of events before note N_{n-1} .

It is interesting to observe that a great number of the various compositional rules of traditional counterpoint and harmony are concerned with zeroth-order Markoff chain effects. Many of the rules of first-species strict counterpoint used to generate the music in Experiments One and Two illustrate this very nicely, particularly the rules forbidding certain types of melodic intervals such as sevenths and tritones. More generally, many rules of conventional harmony are also of this type, for example, the rules given for the resolution of dissonant chords to certain choices among consonant chords or possibly other dissonant chords. It is important to note that in so far as they are usually given, these rules are stated without reference to musical context, this being considered a separate problem.

In Experiment Four, three samples of zeroth-order melodic writing were produced to illustrate this simple type of musical organization. The three unperturbed transition-probability functions shown in Table 15 were used directly for this purpose; thus, the probabilities w_{ij} for selecting interval I_i were simply set equal for all values of i to the values x_j of the harmonic function, y_j of the proximity function, and z_j of the combined functions, respectively, in three successive studies.

First-order Markoff Chain Music. The simplest way to improve upon the structural concept of zeroth-order Markoff chain music is to make the choices of new melodic intervals dependent on what has just immediately happened, that is, to make the choice of interval I_i dependent upon the choice of interval I_{i-1} . This is an example of a first-order Markoff process, and music written according to this principle may be termed *first-order Markoff chain music*. In conventional music, there are examples of first-order Markoff processes among composition rules, just as there are of zeroth-order rules. Among the rules of counterpoint, the skip-stepwise rule and the forbidding of certain types of parallel motion can be cited.

Given the functions in Table 15, the question arose as to how we might use these functions to produce some simple examples of first-order Markoff chain music. From among the many possibilities, the following first-order process was selected: A rule was employed which was a generalization of the skip-stepwise rule previously used. This rule required that the choice of the new interval I_i would be weighted most heavily against the particular interval previously selected, I_{i-1} , and most heavily in favor of the interval most different from interval I_{i-1} . Thus, for example, if the previous interval I_{i-1} had been a melodic repeat, the new set of probabilities for interval I_i would be weighted more heavily against the choice again of a repeat than if the previous interval had been a major third or some other intermediate interval.

Addition is the simplest process for carrying out this type of conditional first-order weighting, utilizing the basic functions in Table 15. It was possible simply to add the weight of the interval I_{i-1} to all the weights of the interval to be chosen, to obtain new weights for the actual choice process. In actual practice, the weights $w_{i-1} - 1$ rather than w_{i-1} were added to w_j . Examples of this process are shown in Table 18, utilizing the harmonic function, x_j . It is seen that the probability for a repeat following a repeat was much less than for a repeat following a tritone. Conversely, the probability of a tritone following a tritone was much less than a tritone following a repeat. In general, therefore, with this treatment of the harmonic function, dissonant melodic intervals tended on the average to follow consonant melodic intervals, and, conversely, consonant melodic intervals tended to follow dissonant ones. A similar treatment of the proximity function y_j would have the effect of causing small intervals to tend to follow large intervals and vice versa, thus providing a generalized sort of skip-stepwise rule.

As in the case of zeroth-order Markoff chain music, three samples of first-order Markoff chain music were generated, again illustrating (1) the effect of the harmonic function alone; (2) the proximity function alone; and (3) the combined function. This function w_{ij} , therefore, was set equal in the three cases to $x_i + x_{i-1} - 1$, $y_i + y_{i-1} - 1$, and $z_i + z_{i-1} - 1$, respectively, according to the calculation pattern illustrated in Table 18.

Higher-order Markoff Chain Music. A second-order Markoff process is produced whenever a new choice is made dependent upon the previous two events; a third-order process arises when the choice depends on the previous three events; and so on. It is possible, therefore, to produce by such simple reasoning complicated interactions between a new event and

Table 18
Examples of How First-order Weighted Transition Probabilities May Be Computed,
Using the Harmonic Function, x_1 , as an Example

Table 18 (*Continued*)

Minor sixth	8	7	Repeat Octave	13 12	20 19	182.	109.9 104.4
Tritone
Major seventh	.	.	.	2 1	9 8	.	49.5 44.0
Tritone
Repeat Octave	0	1	13 12	13 12	12	91	142.9 131.9
Tritone
Major seventh	.	.	.	2 1	2 1	.	22.0 11.0

events that have preceded it. Second-order and higher-order relationships between notes can be found in composition rules. However, a simple cascading process of this type would not produce musical output bearing recognizable relationships to normal musical structures as the relationships between notes become more distant. Therefore, instead of continuing by generating second-order Markoff music, we utilized another more relevant concept, namely, that of defining structurally more important and less important notes. This working principle, adapted from the ideas of musical analysis described earlier, was utilized to produce Markoff chain music in which the notes occurring on strong beats, arbitrarily assuming $\frac{6}{8}$ time, were made to depend on one of the generating functions, while the notes occurring on weak beats were made to depend on another of the generating functions. Four examples of music utilizing this principle were produced by means of the harmonic and proximity functions given in Table 15. The organizing principle of this music is illustrated diagrammatically in Figure 16, where it is seen that two samples each of zeroth-order and first-order Markoff chain music were produced by these more complicated interactions.

In Cases (a) and (c), the harmonic function x_j was used to control the successive choices of strong-beat notes, while the proximity function was used for the weak-beat notes. In this elementary way, we built up simple structures analogous to those suggested by the analysis of conventional musical structures. Specifically, the harmonic function was used as a longer-range structural function to block in larger tonal relationships, while the proximity function was used to provide melodic filler inside these larger units. In the remaining two examples, Cases (b) and (d) in Figure 16, the roles of the two functions were simply reversed. The proximity function, y_j , was now made the more structurally significant in order to provide musical examples in which the neighbor-note, or proximity, relationship was predominant.

Markoff Chain Music and Tonality. In Chapter 2, we noted that tonality in musical composition depends in its simplest form upon a recall of the first tone of a given melodic line. It is seen, therefore, if we compare this concept to the Markoff chain music thus far described, that none of the musical examples can be called tonal by definition, since in none of them was there defined a dependence of note N_i upon note N_1 once i exceeded the order of the Markoff chain process being applied. However, it is possible to define an i th-order process in which the controlled intervals are

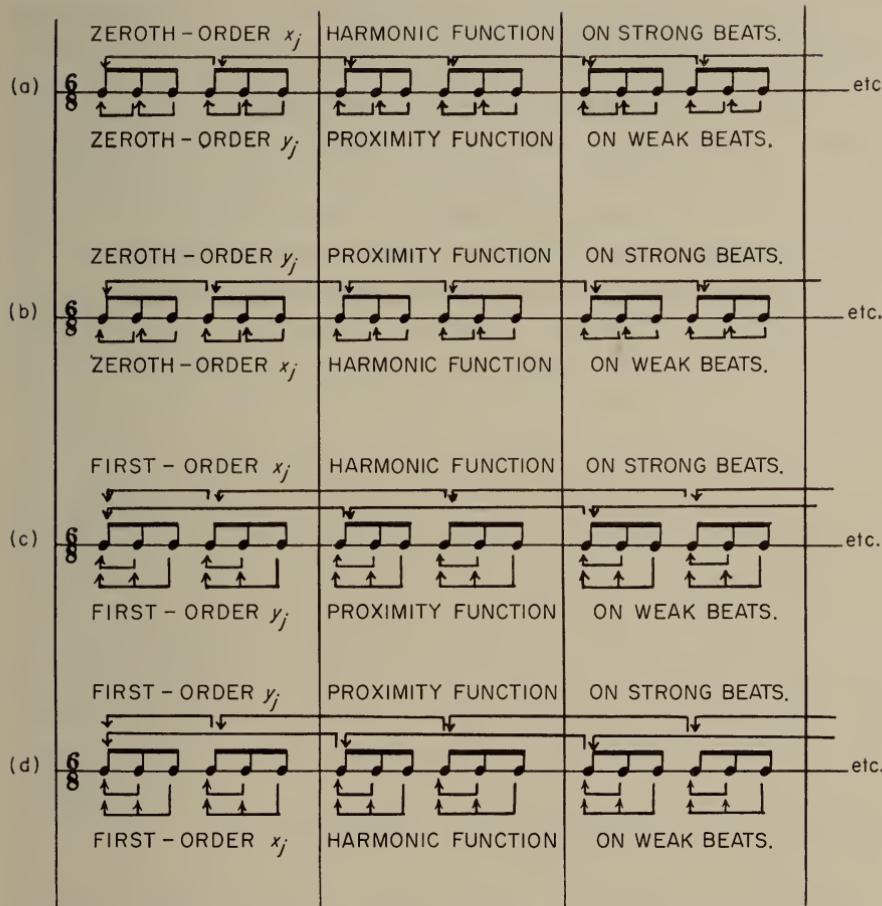


Figure 16. Experiment Four: Illustration of strong- and weak-beat structures utilized to generate species of Markoff chain music.

$N_i - N_1$ rather than $N_i - N_{i-1}$. Once we do this, we relate each new note to the initial note, rather than to the immediately preceding note. We immediately establish, as a consequence, the most elementary type of tonal control in accord with the simple definition of tonality just given. It is interesting also to note that zeroth-order and i th-order Markoff chain music bear a relationship toward one another analogous to that between the interval rows and tone rows discussed in connection with Experiment Three.

Therefore, in the last part of Experiment Four, a sample of i th-order music was produced, but in combination with first-order Markoff music. The harmonic function was used on the strong beats, assuming $\frac{6}{8}$ time, and was applied as an i th-order process. Therefore, each strong beat was related to the initial and, as we shall see, final note of this last sample of music. These notes were arbitrarily set as C in accord with the over-all tonality selected for the *Illiad Suite* assembled out of the musical examples. The weak beats were then assigned a secondary role, one of ornamental or melodic filler. This type of function was best expressed by the simpler type of Markoff process and by means of the proximity function, y_j . The musical pattern, therefore, was the one shown in Figure 17.

ith - ORDER HARMONIC FUNCTION ON STRONG BEATS

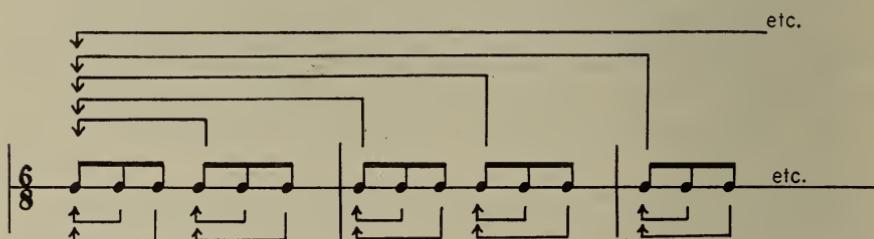


Figure 17. Experiment Four: Musical organization of the last example of Markoff chain music in Experiment Four.

Printout and Modulation. All examples of Markoff chain music were arbitrarily printed in blocks of four lines at a time to permit transcription of the results for the four instruments, Violins I and II, Viola, and Cello. In this way, we were able to provide a maximum amount of musical examples in a minimum of score and playing time.

While writing the printing instructions for the Illiac for this last example, we also incorporated a simple demonstration of how modulation might be handled. Modulation can be defined as a shift of the reference point in an i th-order Markoff chain process to some new tonality. It is possible to design a simple program for resetting the reference point for the i th-order process after any desired number of measures, or, for that matter, after a randomly selected number of measures. Moreover, the new reference point can be preset, or it can be chosen at random, or by some sort of more elaborate, restricted choice process. For example, it might be decided that after

$m - 1$ measures, a modulation from C to F was required. Therefore, on measure m the reference point for the i th-order Markoff chain process would be shifted from C to F. To effect this, it would be possible to rewrite the instructions for generating new notes. However, instead of rewriting the instructions, a simpler method involving the printout routine was employed, because it was evident that modulation is really nothing more than a transposition of the printed results. Thus, to shift from C to F, all we needed to do was to add the number 5 to each note while it was still in the machine as a number. This converted a C to an F upon printing, a C sharp to an F sharp, and so on, and yet did not disturb the basic note-generating process. It is seen that this is effectively what is done by any musician or composer in effecting a transposition or a modulation.

We selected the sequence I-IV-V-I, a simple cadence formula, to illustrate this technique and combined it with the previously described tonal i th-order Markoff chain music to illustrate how modulation, as well as simple tonality, might be achieved. Since C was selected as the fundamental tonality, the subsidiary tonalities became F and G, and the over-all tonal pattern produced became C-F-G-C. The shift to F was set to occur after six measures, the shift to G after another six measures, and the return to C after still another six measures. The use of a preset pattern of this type was preferred this time, since the use of more random processes for instituting changes of this type was amply illustrated by the musical examples in Experimental Three.

Tonal Drive. Modulation is less commonly effected in a piece of music by an abrupt transition from one key to another than by a carefully and logically planned sequence of progressions which give the listener a sense of having arrived at a satisfactory resting point after the modulation is completed. Thus, for example, modulation from the tonic to dominant is a common feature of the exposition section in conventional sonata form. In a simple example of sonata form, the main subject solidly grounded in the tonic key is first fully stated. It is then followed by a modulatory sequence—often consisting of little more than broken chords—which convey the tonal weight from the tonic to the dominant, often through use of chords such as the dominant seventh of the dominant to produce a new resting point upon which the second subject is grounded. This modulatory section, to be convincing, must convey the sense of having prepared for the arrival of the new tonal center.

In general, this type of planning ahead in building up a tonal structure, which applies the concept of "tonal drive," can occur in modulation, as in

the example discussed, and also at other points in a musical structure, such as at the close of a movement on a chosen tonal center. The chords preceding the end, therefore, would be related to the anticipated tonality, and the process in its simplest form would be a reverse i th-order Markoff chain, the reference point being ahead, timewise, rather than behind.

Thus, it was possible also in the last example of Markoff chain music to institute a simple process of tonal drive to a final tonality of C, utilizing a simple modification of the technique previously developed for changing transition probabilities. All we had to do was to reverse the process previously utilized to produce the first example of Markoff chain music as illustrated in Table 15. This process was, therefore, incorporated also into the instructions for this last sample of music. Unity was subtracted from the transition probabilities after every six notes (two measures), so that after twenty-four measures the only choice left became the melodic repeat. However, this technique, of course, was applied to an i th-order process on the strong beats, and, therefore, the only possible point of resolution was on C. A zeroth-order process, by comparison, might end on any tonality. In addition, the same process was applied to the first-order proximity function on the weak beats, so that weak-beat melodic skips were forced to become smaller and smaller on the average as the end of the cadence was approached.

A Simple Closed Structure. The combination of these techniques formed a logical group of musical entities which illustrate how a simple closed musical structure might be generated by the Illiac. Aside from the simple *cantus firmus* setting of Experiments One and Two, and the interval and tone rows of Experiment Three, no closed structures had yet been written to this point. Moreover, the *cantus firmus* settings and interval and tone rows represented really only rudimentary materials, which in most instances would be used for the building up of larger structures.

Therefore, the application of the harmonic function as an i th-order Markoff chain process on the strong beats and the proximity function as a first-order Markoff chain process on the weak beats, along with the process of shifting the tonal reference point after every six measures and the shift of the transition probabilities after every two measures by unit subtraction until only a melodic repeat could occur, permitted the production of a simple closed structure of twenty-six measures, which represented an extended cadence leading to a final close. This closed structure was utilized as a *coda* for the last movement of the *Illiac Suite* to serve as a simple

prototype for building up more complex structures, such as conventional musical forms.

In general, in closing our discussion of Experiment Four, it might be noted that the actual coding of Markoff chain music was extremely simple by comparison with the programming of strict counterpoint. It was a completely efficient process, since there was no attrition resulting from the production of unusable notes; therefore, no try-again subroutine was required. It is entirely possible, however, that the discarding of notes would have to be resorted to once more complex interactions, particularly those involving cross relationships between the voices, were devised. The speed of production of the present examples of Markoff chain music was limited only by the speed with which output tape could be punched by the Illiac.

C H A P T E R S I X

Experimental Results: The Illiac Suite

Introduction

As noted in Chapter 1, the most satisfactory way to present musical ideas is in the form of actual sound. Therefore, as soon as we had planned Experiment One, we also chose a musical medium through which the results might ultimately be presented. The work described in Chapter 5 was sufficiently extensive so that a four-movement suite could be assembled out of the experimental results to parallel exactly the four major experiments. This piece of music, which we entitled the *Illiac Suite* in reference to the computer used for the experiments, is therefore a chronological research record of our experiments.

In assembling this suite, it was desirable to organize the raw results into a playable, relatively coherent whole. Moreover, it was also desirable to minimize the amount of arranging of the materials, so that as much as possible of the musical content of the suite could be said to be computer produced. This we have done, but in addition to the basic choices concerning which experiments were to be carried out, certain additional decisions were required during the transcription of the computer output to a musical me-

dium. These will now be briefly reviewed so that if the score of the *Illiac Suite* in the Appendix is examined, it will be clear what elements in the score result strictly from the operation of the computer and what elements have been inserted or adjusted as a result of practical necessity. The main problems involved in transcribing the experimental results were the following:

1. The first decision was the choice of a musical medium through which the results could ultimately be heard. The use of electronic or other synthetic means was eliminated in our case, since equipment of this type was not available. A conventional instrumental medium was therefore the only other reasonable alternative, but transcription for a keyboard instrument such as the piano was also eliminated because this would introduce the special restriction of having to have the music fit under the hands at the keyboard. Therefore, since, even in Experiment One, the objective was to produce a four-voiced polyphonic texture, the choice of a string quartet medium appeared logical and convenient; the problems in transcription would be minimized. Secondly, a string quartet has a fairly homogeneous timbre, which was desirable from our viewpoint; and lastly, the practical problem of having the music performed appeared less formidable than with larger or more exotic instrumental combinations.

2. Since much more musical output was produced by means of the *Illiac* than could possibly be used, it became necessary in each of the experiments to employ some sort of unbiased screening procedure to select representative musical output. It was important that we select impartially and not on the basis of aesthetic evaluation, since what we required were average rather than superior results. Therefore, various selection processes were employed, such as arbitrarily using every tenth *cantus firmus* setting, or arbitrarily selecting material at the beginning or end of a sample of output, or choosing examples at random by means of a random-integer table.

3. The choice of an over-all structure of a four-movement suite was made. The four movements were entitled Experiments One, Two, Three, and Four, duplicating the plan of the experiments themselves. Moreover, the internal structures of the four movements were planned to correlate with the details of the four experiments. The only self-contained musical structure produced completely by the *Illiac* is the last section of Experiment Four. This was utilized as the Coda of this last movement of the suite.

4. The choice of tempi for the four movements and in the individual parts of the first and third movements was made independently of com-

puter programming, since no programming for tempi was carried out. It should be noted that this is a simple problem, even with the inclusion of provisions for *accelerandi* and *ritardandi*. However, it was decided to select reasonably contrasting tempi for the movements and to defer the study of this musical problem in favor of other, more important projects.

5. It was desirable to transpose Voice 1, assigned to the cello, two octaves downwards relative to the notation utilized for the printed computer output; this was done to place this voice in the most favorable playing range of the cello. For the viola, used for Voice 2, the music was transposed one octave downwards, again for the corresponding reason. Voices 4 and 3 were played untransposed by Violins I and II, with the principal exception being in the first section of Experiment One, where the part for Violin I was transposed one octave upwards. Since the initial procedure for Violin I tended to place the music inordinately high in the playing range of this instrument, this transposition was subsequently abandoned.

6. Some subsidiary random-integer tables were utilized to select dynamic levels in Experiments One and Two and to distribute the *cantus firmi* and two-voiced settings in Experiment One among the instruments. Some doublings in these parts of this movement were also included to provide some dynamic variety. Also, certain restrictions to only two voices playing at a time were made in Experiment Four during transcription. This was done to add some variety to the musical texture of Experiment Four.

7. The meter for each movement or section of a movement was selected during transcription. In terms of computer programming, this problem is a simple one like programming tempo, and was deferred in favor of other projects.

Outside of several other quite minor additions, the rest of the content of the suite can be considered directly the result of computer programming. The general structural outline of the *Illiac Suite* is shown in Figure 18. The structural details of each movement of the suite will now be considered.

Experiment One

This movement is divided into three sections—*presto*, *andante*, and *allegro*—which contain successively samples of computer output which illustrate how we progressed from the writing of simple *cantus firmi* to two-part and finally four-part settings. In order to show a number of examples of each type of computer output, series of settings of different

Experiment One	Monody; 5 samples each of <i>cantus firmi</i> 3 to 12 notes long.	Two-part <i>cantus firmus</i> settings.	Four-part <i>cantus firmus</i> settings.	
	<i>Presto</i>	<i>Andante</i>	<i>Allegro</i>	
Experiment Two	Four-part counterpoint; from random white-note music to strict counterpoint with rules added successively. Two samples of each type setting. <i>Adagio, ma non troppo lento</i>			<i>CODA</i> Extra samples of cadences.
Experiment Three	Basic rhythm, dynamics, and instrumentation code. <i>A</i>	Random chromatic music. <i>B</i>	Modified rhythm, dynamics, and instrumentation code plus random chromatic music. <i>A'</i>	Revised rhythm, dynamics, and instrumentation code plus random chromatic music. <i>A''</i>
	<i>Allegro vivace</i>	<i>Adagio</i>	<i>Allegro vivace</i>	<i>Adagio</i>
				<i>Allegro vivace</i>
Experiment Four	Alterations of harmonic-function transition probabilities. <i>Tanto presto che possibile</i>	Zeroth-order Markoff chain music.	First-order Markoff chain music.	Separation of strong and weak beats.
				<i>CODA</i> ith-order Markoff chain music; modulation and simple closed structure.

Figure 18. Structure of the *Illiad Suite*.

lengths were strung together to form larger musical structures. Since these first programs for the computer were designed to produce fifty samples each of *cantus firmi* or *cantus firmus* settings from three to twelve notes long, the structures of the three parts of this first movement were designed to exhibit a sampling of all the output of the various different lengths. In the *presto* part of Experiment One, five samples each of *cantus firmi* from three notes to twelve notes long were used successively to build up this section of the movement. To provide reasonably unbiased sampling, every tenth sample of computer output was chosen arbitrarily for inclusion in the suite. These are played individually by the four instruments until near the end, where some doublings are permitted. The distribution of the *cantus firmi* among the four instruments was decided by means of a four-choice random-integer table, also generated by means of the Illiac. The scoring was carried out by associating each instrument with one of the numbers employed to build up the table. Successive integers in random sequence were used one at a time to assign *cantus firmi* to the instruments. It is interesting to note that this seems an entirely adequate way of scoring this type of musical material. The selection of dynamic markings was carried out by a similar technique. This time, the random integers were associated with the markings *ff*, *f*, *p*, and *pp*. Finally, as previously explained, octave transpositions were employed to place the *cantus firmi* into effective playing ranges for the instruments. At the close of this section of Experiment One, a *pizzicato* chord was inserted to show the transpositions used during transcription.

The second section of this movement (*andante*) is a sampling of two-part *cantus firmus* settings. Here we put together successively individual samples of three- to twelve-note settings and then reversed the order to provide a change of structure and, hence, some musical contrast to the first part of the movement. The selection of parts and dynamics was carried out as in the first part of the movement, utilizing the same four-choice random-integer table, except that, for the choice of the instruments, the integers were used two at a time.

In the final section of the movement (*allegro*), a group of four-part *cantus firmus* settings of increasing length make up the structure, repeating the plan of the first part of the movement. Dynamic markings were chosen as before. The movement ends with an arbitrarily chosen sample of the settings, in which we allowed the voices to hold notes over many beats in violation of the rule against successive repeats. Also, a final *pizzicato*

chord, duplicating the ending of the first part of Experiment One, was inserted for structural symmetry.

It is evident, upon examination of the score of Experiment One contained in the complete score in the Appendix, that in these settings the rules employed for Experiment One are not violated. However, it should be kept in mind that these various *cantus firmi* and *cantus firmus* settings exhibit faults one might naturally expect as a result of the absence of the remaining strict counterpoint rules. This is not to say that many are not without some musical interest within the limited technique given the computer, but only that the strict counterpoint problem was still incompletely solved. Obvious musical faults, in terms of strict counterpoint, include the outlining of chords such as triads, sequences of melodic skips, six-four chords, and a number of other imperfections. For reasons previously mentioned, it was decided to complete our study of the strict counterpoint problem by writing improved settings of the type included in Experiment Two.

The musical content of Experiment One, of course, has other obvious limitations. Aside from the extreme simplicity of the melodic and harmonic style, perhaps the most important limitation is the absence of harmonic change. The whole movement rests squarely on a C-major tonality, which yields a rather unusual static but not necessarily unpleasant effect. An alternate procedure, which could have been carried out even at this early point, would have been to score the *cantus firmus* settings in different keys chosen perhaps randomly or by some predetermined scheme. However, since this would have been an added musical element not directly related to the main objective of Experiment One, the use of such a device was not considered.

Experiment Two

In this movement, as previously explained, we wished not only to show that we can write essentially correct first-species strict counterpoint by means of a computer, but also how the imposition of counterpoint rules leads to a clarification of the texture of purely random white-note music. Therefore, in Experiment Two, the *adagio* of the *Illiad Suite*, we have assembled a sequence of *cantus firmus* settings, all arbitrarily fixed at a length of twelve notes. After starting with two samples of purely random music, as each section of the movement is reached, the number of rules

governing the choice of the notes is increased in accord with the plan shown in Table 19. The movement ends with four extra representative

Table 19
Experiment Two
Sequence in Which Strict Counterpoint Rules Were
Successively Added to Random White-note Music

<i>Section</i>	<i>Added rules</i>
[A]	Random music; no rules
[B]	Skip-stepwise rule; no more than one repeated note
[C]	<i>Cantus firmus</i> starts on C with C chord for opening; cadence on C with leading tone in one of the four voices; resolution of tritone in VII ₆ , e.g., F must resolve to E B must resolve to C
[D]	Octave-range rule
[E]	Only consonant chords permitted except for $\frac{6}{4}$ chords; i.e., harmonic subroutine added
[F]	Parallel unisons, octaves, fifths, and fourths still permitted; melodic subroutine added
[G]	Parallel fourths; $\frac{6}{4}$ chords containing tenth still permitted
[H]	Best counterpoint

cadences written by means of our cadence program. Throughout, as in the first movement, dynamics were chosen by means of a four-choice random-integer table, and the over-all structure and octave transpositions were worked out during the course of transcription.

Two passages from Experiment Two might be given special consideration in examining this music, namely, the random white-note music used to open the movement and the highly organized music used at the end of the movement. A comparison of these settings illustrates how order can be imposed upon random music by the logical processes described in detail in Chapter 5. Experiment Two, therefore, is a simple musical illustration of how the introduction of redundancy into a structure with a relatively high entropy content brings about a clarification of texture. Since the settings also became progressively more difficult for the computer to work out, this experiment also shows how redundancy reduces the information which may be communicated and how increasing the redundancy can only be brought about by increasing the amount of material which must be rejected.

This movement happens also to be one elementary example of how a high-speed digital computer can be used to experiment with musical prob-

lems in a novel way, since the writing of a sequence of textures in the manner illustrated would be difficult to do by conventional means. The composer unaided by a computer is not normally conditioned to maintain an unbiased attitude toward his output, so while he could write the random white-note music with the aid of a random-number table easily enough, beyond this point it would be difficult for him to cope with the problem of obeying only certain rules and letting everything else occur at random.

Experiment Three

Each of the three long fast sections of this movement is built upon the rhythm, dynamics, and playing-instructions codes described in Chapter 5. The first part of the movement is an example of how the simple version of these codes operates. The only addition to the computer output here is the insertion of pitch levels for the four instruments. These were selected by means of an arbitrarily chosen tone row, in anticipation of the Coda of Experiment Three in which computer-generated tone rows are presented. All other elements in the opening measures of Experiment Three are seen to be based upon computer output of the type shown in Figure 9. The output for this section was transcribed directly as produced by the computer, since no selection process was necessary in this experiment.

The first *adagio*, section [C] in the score, for which both the $\frac{3}{2}$ meter and *ff* dynamic level were selected during transcription in addition to the slow tempo, is an example of random chromatic writing. This is music of the highest possible entropy content in terms of note selection based upon the normal chromatic scale. It has a higher entropy content than the white-note random music of Experiment Two, since there are five more available choices per octave, namely, the black notes. This is truly atonal music of the most extreme sort, as we have noted already in Chapter 5.

In the second *allegro vivace* section of Experiment Three, this random chromatic writing is combined with the second, more complex rhythm, dynamics, and playing-instructions code to produce a rather highly complex dissonant musical texture. In combining these two codes together, it should be noted that only on the 1s, or "strikes," were the notes of the random music utilized. On the 0s, signifying "hold" or "rest," the notes were skipped. In this section, each random note produced by the computer was denoted rhythmically by an eighth note.

The second *adagio* section, section [G] in the score, illustrates what happens when the few simple rules of writing described in Chapter 5 are imposed upon random chromatic music. It is seen that this music is much more controlled than the music in the earlier *adagio* section. These results seemed to bear out our anticipation that the particular rules of writing selected would be efficient in imposing order upon the musical output.

In the third *allegro vivace*, this type of material is combined with the second rhythm, dynamics, and playing-instructions code. The same technique of combination was employed as used in the previous *allegro vivace*. This music, in some ways, resembles certain kinds of contemporary music. The continually dissonant but rather colorful musical texture, the use of complex rhythmic *ostinati*, the shifting degrees of rhythmic and dynamic independence of the four voices, the absence of conventional tonality as a structural device, and the nature of the melodic profile all suggest elements of twentieth-century style. Aside from crucial factors such as tonality, the major element missing is perhaps the use of imitation and melodic repetition as a structural device. Because of this, we started at one point writing a computer program for composing simple fugues in this general style but later set aside this project in favor of the studies in Experiment Four. It might also be noted that our choice of the tritone resolution as our one element of harmonic control happens to reflect Hindemith's definition of two basic types of chords—those without tritone and those with tritone.¹ The two chord types in Hindemith's system bear different structural implications. Our choice of this device happened to be arrived at independently and came in part from our experience with the counterpoint experiments. However, it is interesting to note in Hindemith's writing a similar emphasis on the importance of the tritone in structural harmony.

One additional point struck us immediately when we compared these results from Experiment Three with the output produced in Experiment Two. In these two experiments, we have contrasted two widely different styles—one bearing a relationship to sixteenth-century musical style, the other to twentieth-century style. One style is highly restrictive, highly consonant, but sounds quite simple, while the other style sounds dissonant and much more complex and difficult to decipher. It is important to note, however, that simplicity of style and hence accessibility bears an *inverse* rela-

¹ P. Hindemith, *The Craft of Musical Composition. I. Theory*, rev. ed., Associated Music Publishers, Inc., New York, 1945, pp. 94–108 in particular.

tionship to the freedom of choice. The simplest style requires the severest restrictions and has the highest degree of redundancy. On the other hand, simpler musical styles are by no means necessarily the easiest to write, since the difficulty of composition involved in making the best choices from among the many available in larger structures in a less restricted style is offset by the fact that more of the available choices are permissible in terms of the desired end effect. This is illustrated by the comparison of Experiments Two and Three. The work required to extract the type of order illustrated by first-species strict counterpoint was greater than that required to produce the last *allegro vivace* of Experiment Three; the programming was more complex and more difficult. For a structure of given dimensions, we may therefore suggest that, as a general rule, a more restricted style is more difficult to write, since fewer of the interactions between the notes can be left to chance; or conversely, more of the interactions must be precisely defined. This conclusion is compatible with a composer's experience in writing a highly restrictive sample of music, such as a counterpoint setting, in comparison with the writing of a more loosely organized piece of music in which the relationships are permitted to be much more casual.

The Coda of Experiment Three, section [K], consists of one example each of an interval row, of a tone row, and of the restricted type of tone row described in Chapter 5. Each row utilized was arbitrarily assigned as a solo to the cello and followed by the inversion, retrograde inversion, and retrograde forms of the row played on the remaining three instruments. As described in Chapter 5, these transformations of the row were easily produced in the computer by changes in the printing instructions in the computer code. The code written to produce these rows could, of course, be utilized in future experiments as a basic subroutine in programs to produce more complex serial compositions.

In reviewing the musical results of Experiment Three, we have felt that one point at which the results might have been easily improved is in the repetition pattern of the rhythm, dynamics, and playing-instructions code. Setting the index for this at a maximum of twelve measures made the music seem to us somewhat overly repetitive. Resetting this factor involves changing only one number in the set of instructions for the computer. This, of course, is a very simple matter. An index of about six measures rather than twelve would perhaps be more nearly the desirable value to select.

Experiment Four

In preparing the score of Experiment Four, the experimental results were transcribed directly as produced by the computer, since all samples of output were equally good. Fast tempo and $\frac{6}{8}$ meter were arbitrarily chosen to provide a contrast to the previous movements, and dynamics indications were inserted during transcription independently of computer programming. As pointed out previously, results were transcribed for all voices simultaneously to provide more experimental material in the score in playable form. These experiments in monody were transcribed, therefore, as pseudopolyphony. Indeed, it might even be noted that a compositional technique of this type has historical precedents of various sorts; for example, in the part writing in late medieval music.²

The first two sections of this movement illustrate how changes in the transition probabilities derived from the harmonic function change the character of the music. In the first sample of zeroth-order Markoff chain music at the opening of the movement, the transition probabilities are adjusted after every two measures, so that by the time this initial section is completed, a sample of every distribution from only the repeat being permitted to the so-called average distribution has been illustrated. This is in accord with the shifts in transition probabilities shown at the far left-hand side of Table 15 in Chapter 5. The net effect of these shifts is to cause the music to progress from a fixed C tonality to a rather free texture in which the statistical distribution of melodic intervals is in accord with the "average" or unperturbed distribution. In examining the transcription of these results, it should be noted that each two measures represent the first sample of a larger amount of computer output representing each type of distribution. Therefore, across the bar lines of every other measure, the rules are not observed.

The second group of results, section [A] of the score, is a similar series

² Thus, "Franco [of Cologne] states that whoever wishes to write a conductus should first compose his own tenor . . . and then add the *discantus* (i.e., the du-plum); if a third voice is wanted, care should be taken always to have it in concordance with either the tenor or the *discantus* . . . indicating that . . . the parts were written one at a time, rather than simultaneously." (G. Reese, *Music in the Middle Ages*, W. W. Norton & Company, Inc., New York, 1940, p. 308). This type of construction, as a matter of fact, might be of considerable interest because it could serve as a connecting link between the Markoff chain monody produced in the present Experiment Four and full-fledged polyphony.

of extracts from computer output in which a progression from random distribution, that is, equal occurrence, of melodic intervals to the unperturbed distribution is illustrated. In this example of zeroth-order Markoff chain music, we progress, therefore, from a more random to more restricted situation as opposed to the opening part of the music.

The next parts of Experiment Four contain zeroth-order Markoff chain music first based upon the harmonic function above (section [B] of the score), then upon the proximity function alone (section [C] of the score), and lastly upon the combined function (section [D] of the score). This is followed immediately by a similar sequence of samples of first-order Markoff chain music (sections [E], [F], and [G] of the score) which serve structurally in the movement as an answering section. In the presentation of these samples of computer output, as noted previously, the number of voices playing at one time was reduced arbitrarily from four to two, except for the playing of the combined functions. This was done simply to provide some additional internal dynamic variety to this last movement. It is observed that the character of each of these samples of music is in general accord with the rules which govern its production. The harmonic function music is characterized by a preponderance of skips and broken-chord patterns; octaves, fifths, and other consonant harmonic intervals occur much more frequently than stepwise motions. On the other hand, the proximity-function music contains a higher proportion of stepwise intervals, neighbor-note motions, and fragments of scales. In general, the over-all character of this music is more melodic in texture. Finally, when the additive combined function is utilized in a first-order process, the nearest approach to recognizable melodic writing possible through utilization of this general technique seems to have been reached. The first-order process tends to promote a type of skip-stepwise rule, as previously explained, so that wide skips tend somewhat more than in the zeroth-order process to be followed by repeats and stepwise motions and consonant leaps to be followed by dissonant leaps, and vice versa.

The next several short sections of the movement contain examples of music in which the weak- and strong-beat functions have been differentiated, firstly in a zeroth-order process and then in a first-order process. Even within the limited set of results kept in the score, the operation of the composing scheme outlined by means of Figure 16 in Chapter 5 can be distinguished. Taking the two contrasting zeroth-order samples of music from sections [H] and [I] of the score of Experiment Four, we see upon extraction of the strong-beat notes from these samples, as shown in Figure

19, that the sense of the harmonic function predominates in example (a), while the sense of the proximity function predominates in the example (b). Moreover, an inspection of the complete examples in sections [H] and [I] reveals that the weak-beat notes have been selected in general accord with the computer instructions.

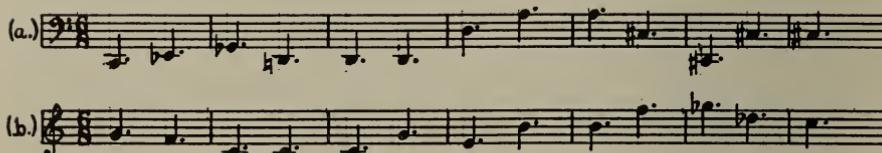


Figure 19. Experiment Four: Extraction of the strong-beat notes from samples of zeroth-order Markoff chain music in sections [H] and [I] of the full score.

The Coda of Experiment Four appears to disclose in a satisfactory way all the features expected of it as described in Chapter 5. The modulations worked out as planned, and the harmonic and melodic simplification develops measure by measure as planned until the end of the movement is approached. Toward the end of the movement, on the strong beats, only the notes closely related harmonically were permitted to occur, while on the weak beats, only neighboring notes were permitted, until at the very end only the tone C was allowed. All of this worked out in accord with the computer programming and indicates that the design of more complex closed musical structures might be started by using this simple prototype as a point of departure.

CHAPTER SEVEN

Some Future Musical Applications

Introduction

It is convenient to define two general areas of interest which might be considered in applying computers to musical problems in the future. One of these we may describe as the theorist's field of interest; the other, as the composer's. The theorist analyzes music written by composers to characterize musical forms and how they operate. On the other hand, the composer is more of an empiricist who seeks out new musical forms which seem to him to be satisfactory. In attempting to suggest how various new musical projects might be started from where we have left off, we shall arbitrarily group our suggestions into these two basic categories—the first related in general to the analyst's problems, the second to the composer's.

Applications to Music Analysis

The fundamental role of the music analyst is to verbalize specific musical problems so that aspects of musical communication can in turn be defined.

In essence, the problem for the musical analyst is to find the reasons why a composer accepts or rejects musical materials, and to this end, principles of musical aesthetics such as those reviewed in Chapter 2 have been gradually established. We have indicated, however, that much of the writing on aesthetics is not yet particularly precise—Langer's "significant form" is a case in point. We have also indicated that the investigation of specific forms, rather than general concepts, is to be preferred for the time being in seeking more precise definitions of musical concepts. This proposition follows from the argument of significant overlap of levels of communication proposed by Weaver, which we referred to in Chapter 2. The description of musical forms perhaps ultimately can be expressed in terms of information theory or some future equivalent, although, at the present time, application of this theory to musical analysis is perhaps still too recent to evaluate critically in any detail.

If we now tabulate various applications within this general field of interest, the following suggestions come to mind as representative examples of the large number of possible projects which might be carried out:

1. Perhaps the most obvious application of computers to musical analysis is the extension of the type of studies already illustrated by the *Illiad Suite*, in which we have applied the Monte Carlo method to the problem of musical form. As a consequence of coding aspects of this problem as numerical information and generating experimental results by means of a computer, a computer is made to behave as a specialized, but unbiased composing apparatus existing in a completely isolated environment, subject only to the controls and information the music analyst might wish to supply. In this application, a computer is an ideal instrument by means of which analytical ideas can be tested, since the investigator starts with certain hypotheses from which he formulates operating principles; he supplies this information to the computer; the computer then generates music based upon these principles; and the investigator then analyzes the results to further his investigation. This, of course, is essentially nothing more than a standard example of experimental scientific method, but the unusual thing is that computers provide a practical experimental technique for carrying out such research in the musical field. It can reasonably be assumed that in the future the combination of these techniques with the more purely theoretical and speculative studies in the musical field, such as those by Moles, referred to in Chapter 2, would be profitable.

In our work, Experiment Four, in particular, suggests a technique of how a new type of analysis of musical forms can be brought about. This

experiment, the theoretical basis of which is contained in information theory and more generally in statistical theory, could only be carried out in a reasonable time with the aid of devices such as the Illiac. As an initial working premise for future studies of this type, it is suggested—with appropriate restrictions—that most musical compositions reflect a balance between the extremes of *order* and *disorder*, and that stylistic differences depend to a considerable extent upon fluctuations relative to these two poles. As a second working premise, it is suggested that interest in musical structures is achieved normally when fluctuations around some stylistic mean between these two poles are also subject to processes of organization and arrangement timewise within the structure. The testing of these two general premises through the technique of generating Markoff chain music would be a logical extension of the work of this type thus far completed.

2. In addition to these more general studies, there are many specific tasks of musical analysis that could also be carried out with the aid of computers. The analysis of specific musical styles following perhaps some of the methods cited on pages 133 and 134 is one example. For example, estimates of the relative degrees of order and disorder of different samples of music or different sections of given musical structures could be attempted. This is suggested since entropy seems to be a more useful variable than less well-defined concepts such as "harmonic tension." Characteristic melodic profiles for different styles could also be examined and codified. Studies of this sort would be of particular interest in musicological research, such as finding the determining characteristics of particular styles of historical interest. It is also possible that the results of such analysis could be used in a practical way to identify, to sort, and to catalogue old music—often a tedious and laborious task. Thus, for example, it might be possible to determine whether samples of music merely represent variants of one basic piece or fundamentally different pieces. With adequate analytical criteria, at least a major part of such a problem could be coded for a computer to permit automatic cross comparisons of the samples and subsequent sorting into appropriate categories. As a specific example, at the present time we are considering a project for transcribing, sorting, and reproducing French lute music in a complete modern edition.¹

¹ D. Lumsden, *Un catalogue international des sources de la musique pour luth (Les leçons d'une étude des sources anglaises)*, CNRS colloque, "Le Luth et sa musique," Paris, Sept. 14, 1957; T. E. Binkley, letter to Jean Jacquot, President of CNRS, Paris, based upon comments upon Lumsden paper just referred to.

Since there is a vast quantity of this music, it has been estimated that up to ten years will be required to do the job by ordinary means. It has been suggested, however, that instruments such as the Illiac could be used to speed up the process. It should be pointed out in connection with this problem that a similar application of computers has already been made in the field of Biblical research and also in the preparation of a concordance for the new revised standard version of the Bible.² It has been pointed out that it took thirty years to prepare the concordance for the St. James version of the Bible, while for the newer Bible, the same task was carried out in nine months with the help of a computer.

It should be stressed in this connection that such a project would include the automatic conversion of old music into modern notation in score form, along with performance parts, if desired. Utilizing a computer, this older music could be worked up rather easily and after final editing could be made available in modern copy. The first step in carrying out this project is the design and construction of a suitable musical printout mechanism of the types discussed in Chapter 4. Secondly, the most efficient utilization of computers for such purposes would depend on the development of adequate scanning devices for computers which would recognize printed and even handwritten letters and numbers. At the present time, research is being carried on to produce such scanning devices. This was given recognition in a recent symposium devoted to document reading, pattern recognition, and character synthesis.³ When these devices become generally available, it should be possible to adapt them to the scanning of musical notation. In the meanwhile, transcription of musical materials onto coded tape or punched cards is an adequate substitute technique.

Other possible practical applications in musicology might include the use of computers to realize *continuo* and figured bass in Baroque music and to complete the part writing in older music where the music has either been left incomplete or some of the parts are missing. In this last application, as a result of informed statistical style analysis, at least a highly probable realization of the missing parts could be produced.

3. Another practical application in this general area, namely, to peda-

² A. Carpenter, "Amazing New Uses for Robot Brains," *Sci. Digest*, 41(2):1, February, 1957.

³ L. Cohn, R. A. Kirsch, L. C. Ray, and G. H. Urban, "Experimental Use of Electronic Computers in Processing Pictorial Information"; T. L. Dimond, "Devices for Reading Handwritten Characters"; A. I. Tersoff, "Automatic Registration of High-speed Character-sensing Equipment," all given at Session IX of the 1957 Eastern Joint Computer Conference, Washington, D.C., Dec. 9-13, 1957.

gogical uses, should be mentioned. Several such applications have been suggested to the authors. On pages 128 and 129 the use of restricted tone rows as *cantus firmi* for counterpoint studies was discussed. Extensions of this work, such as the generation and cataloguing of such tone rows into different groups with characteristic properties, has been proposed by Robert Kelly, the author of the counterpoint method for which these tone rows are intended.⁴

The systematic generation of musical materials for teaching manuals for instrumental performance would be a second application of this type. The preparation of manuals for the guitar and the lute has also been proposed.⁵ In this application, the coding would be based on guitar and lute tablature rather than standard musical notation, since tablature is in itself already a codification of the technical limitations of these instruments.

4. A final application to music analysis, we should like to mention, is the analysis of musical sounds themselves. This type of information would be particularly useful in the production of synthetic music by means of computers in conjunction with other electronic equipment. A very considerable knowledge of musical sounds and their physical constitution is, of course, available today.⁶ Moreover, a large amount of information on how to process these sounds by means of electrical and electronic equipment has been built up through the development of sound-reproduction systems, broadcasting, and other commercial developments, to say nothing of the more experimental techniques reviewed in Chapter 3.

Musical sounds are, of course, extremely varied, and the analysis of timbre, attack, and other factors which make up these sounds becomes quite complex and is by no means completely understood. It seems possible that a computer might be useful in improving the design of sound-producing equipment in one significant way in particular. The analysis of sound could be coded as digital information, using standard techniques such as Fourier analysis and the like, and stored for computer utilization in permanent form, perhaps, on magnetic tape or equivalent high-capacity storage. Instruments such as the RCA Synthesizer might be extremely use-

⁴ R. Kelly, private communication.

⁵ T. E. Binkley, private communication.

⁶ Typical books on musical acoustics are numerous and include H. L. M. Helmholtz, *On the Sensations of Tone*, Dover Publications, New York, 1954; Alexander Wood, *The Physics of Music*, The Sherwood Press, Cleveland, Ohio, 1944; C. A. Culver, *Musical Acoustics*, 4th ed., McGraw-Hill Book Company, Inc., New York, 1956; L. S. Lloyd, *Music and Sound*, Oxford University Press, New York, 1937; H. F. Olson, *Musical Engineering*, McGraw-Hill Book Company, Inc., New York, 1952.

ful as primary sources for acquiring such stored information, or, alternatively, the analysis of actual sounds might be carried out in a computer by reversing the flow of information through a digital-to-analog type device of the kind we will discuss in the next section. Basic programs for extracting this information out of storage and building up complex sound structures could also be developed. These synthesized sound patterns could be printed out as digital results and used to process sound by means of instruments such as the Synthesizer. Or, more directly, this information could be reconverted directly to sound, using the digital-to-analog conversion units of the type discussed below.

Applications to Music Composition

1. There are many practical problems of composition which require examination in order that the rather limited catalogue of compositional techniques thus far treated might be extended. We may note, therefore, just a few of the more urgent of these problems to illustrate the nature of what could be done: (a) the writing of computer programs for handling many of the traditional and contemporary harmonic practices; (b) the writing of more complex counterpoint programs, including programs for more contemporary counterpoint; (c) the correlation of elements such as rhythms and dynamics to note selection; (d) the use of imitation as a structural device; (e) the use of thematic and melodic recall and development; (f) the coding of factors thus far neglected, such as tempo, meter, and choice of instruments; (g) the writing of standard closed forms, such as variation form, fugue, song form, sonata form, etc. This last is of obvious major importance. Not only specific forms, but the more general problem of form needs to be considered here. The application of ideas such as Schenker's concept of chord prolongation and of Meyer's concept of purposeful musical motion would undoubtedly be useful in these studies, to say nothing of the possible results of analytical studies such as those proposed in the previous section. In time, it is conceivable that the handling of many elements involved in the writing of standard musical textures might be carried out relatively simply and efficiently with a computer. This would depend, of course, on the ease and precision with which these musical elements could be programmed and the information stored in sufficiently compact form. The extent to which this may become possible is, of course, a matter for speculation at present, so we can only suggest that music-writing programs might be designed to produce music efficiently

by utilizing, among other things, standardized "library subroutines" for standard musical operations, much as today standard subroutines are utilized for ordinary mathematical operations. In this connection, it is worthwhile to note that the writing of the computer programs themselves may very well be made more efficient. Since the time consumed in computer programming normally requires much more time than the actual time of computation in a computer, there is intensive research today into the possibility of computers themselves generating detailed computer programs from more general sets of instructions.⁷ Therefore, if these new techniques are developed, there is no reason to suppose that they could not be adapted to the coding of musical as well as mathematical problems.

2. The organization of more or less standard musical materials in relatively novel musical textures, including combinations not easy or even feasible by other means, might be carried out. Some possibilities which come readily to mind include: (a) the use of different rules or even of different styles between different voices in a polyphonic texture; (b) the inversion of rules, forbidding what is now permitted, permitting what is now forbidden; (c) the development of new rules of operation for handling musical materials, such as subjecting tone rows to complex permutations based upon the concept of these rows as arrays spaced across 12×12 unit plots of pitch versus time. A project of this type could perhaps very well be founded on some of the permutterational composition techniques referred to in Chapter 3. In this context, the totally organized music, also referred to in Chapter 3, would appear to be a composing technique particularly suitable for computer processing.

3. There might be developed sets of new organizing principles for musical elements leading to basically new musical forms. In the *Illiad Suite*, we have already provided a number of specific examples of how this can be done. The production of random musical elements, either notes, or rhythms, or scoring, is one example. This represents the opposite condition of totally organized music and, as such, becomes a formal element to be integrated into musical structures. More generally, the control of a precise degree of randomness and of the fluctuation of musical texture between order and disorder would seem to be more easily controlled by computer processing than by other means. Obviously, this development of a composing style consciously based upon this picture of musical structure could be of significance in developing an aesthetic not only related

⁷ For example, see D. D. McCracken, *Digital Computer Programming*, John Wiley & Sons, Inc., New York, 1957, chap. 18 in particular.

to concepts of information theory, on the one hand, but also, on the other hand, of more general significance than such relatively restricted concepts as traditional harmonic music, the tone-row technique, totally organized music, and so on. Moreover, since the codes used for the Illiac are based in part upon random-number processes, these codes permit the computer a conditional sort of "freedom of choice," this "freedom" being the equivalent of randomness. The extent of this freedom could be made to depend on how much the aesthetics of music might be expressed in the most general terms. If a large number of highly specific rules, such as the rules of strict counterpoint, are given the machine, the freedom of choice for the computer to select musical materials is quite limited. Not only is the over-all aesthetic quality highly predictable, but the specific details of the music are rarely surprising. This is equivalent to saying that the redundancy is large. However, if a computer is supplied with less restrictive rules, then neither the general aesthetic effect nor the specific musical results are necessarily so predictable. One possible consequence of this is that the composer might no longer be preoccupied with selecting specific notes of the scale, specific rhythms, and other such details, but rather with more generalized sets of symbols, which would be used in turn for the generation and arrangement of the musical details in accord with the musical "message." This could result in a very different attitude toward specific musical details, and they might cease to have the importance we now attach to them. In fact, these details could easily be varied over wide limits within the same essential "composition," just as we now permit, to a greater or lesser extent, variability in the interpretation of music by performing musicians and yet recognize that this does not destroy the uniqueness of a musical composition. It is only fair to note, however, that if such a development were to occur, it would be a radical departure from the attitude prevailing at the present time.

Among the few experiments of this type not involving computers that seem to have been carried out recently are compositions of students of John Cage, such as Morton Feldman, who has written a composition entitled *Intersection No. 3, for Strings, Woodwinds, and Solo Cello*. In this work, Feldman permits a high degree of improvisatory choice by the performers, since the "score" is set down upon graph paper rather than in conventional notation.⁸ A somewhat similar, but less extreme experiment has also recently been carried out by Gunther Schuller in a string quartet

⁸ H. Cowell, "Current Chronicle, New York," *The Musical Quarterly*, 38:123-136, 1952.

composed for the 1957 Contemporary Arts Festival at the University of Illinois.⁹ Thus far, in our experiments, this particular aspect of musical composition has only been barely examined in direct terms. The most conscious application of the idea of controlling the degree of chance events occurring in a musical structure was set up in Experiment Two.

Of perhaps more immediate interest is the further extension of new musical organizing principles related to the general concepts of information theory, such as the Markoff chain music in Experiment Four. More experiments along the line of Experiment Four could be carried out, particularly to develop cross relationships between the voices. It might be possible to start, as previously suggested, with a musical form in which one voice is defined as a *quasi-cantus firmus* to which the other voices are related. Another possibility might be to make direct use or modifications of the techniques suggested by Hindemith¹⁰ by means of which he classifies all chords into six basic groups and each basic group into subgroups depending on the tonal strength of each particular type of chord. This provides a basis for setting up a stochastic variable for the tonal value of vertical note combinations. In turn, moreover, as successive chords are considered, this variable might be used to control the "harmonic tension" of the music as it develops in time and, more importantly, the incorporation of harmonic factors into musical structural units. Hindemith's system has the virtue of being simple and yet amenable to the type of musical propagation developed for Experiment Four, since it is particularly applied to what we have termed zeroth- and first-order chain processes. His ranking of the chords, which seems reasonable as a zeroth-order evaluation, could be valuable as an initial reference standard. Any other zeroth-order basis for grouping chords could, of course, be used just as well, depending on the type of experiments the coder had in mind.

4. Another project might be the systematic study of microtone music. In the past, because of the difficulties of understanding and building systematic harmonic relationships in microtone systems such as quarter-tone music and complex tunings employing microtone intervals to secure just intonation in all keys, and performance difficulties, this field has largely been neglected. Moreover, in the few examples of microtone music we have heard, the smaller intervallic movements seem to fulfill a coloristic rather than a functional purpose. The use of quarter-tones in Bartók's *Violin*

⁹ W. S. Goldthwaite, "Current Chronicle, Urbana, Illinois," *The Musical Quarterly*, 43:390, 1957.

¹⁰ Hindemith, *op. cit.*, pp. 106-108, 115-121.

Concerto and in Julian Carillo's *Preludio a Cristobol Colombo*, to cite two examples, seem to be cases in point. With suitable sound production means, however, a systematic study of the harmonic and contrapuntal relationships in microtone music could be carried out by means of computers. Similar studies could be carried out also on unusual scales and tuning systems, or even variable tuning systems in which tunings could be changed during the course of "performance." Variable tuning, of course, is the ideal technique for securing just intonation in all keys.

5. Perhaps the most significant application of computers would be the combination of computers with synthetic electronic and tape music of the various types reviewed in Chapter 3. This obviously is a natural and complimentary combination, since with computer music we are primarily concerned with organizing musical materials, while with synthetic music we are concerned more directly with the production of sound. Certain specific rather immediate results can be predicted. In future experimentation with computer music, the advantages of being able to produce the results directly in sound as well as in score form are obvious. Not only would the results be analyzed more efficiently, but the means would be available for producing quickly and efficiently the final desired musical end result. Moreover, in view of certain of the other projects outlined above, the experimentation would no longer have to be confined to musical materials for conventional scales, tunings, and instruments.

However, it is not only to the advantage of future experimentation with computer music that the combination of techniques should prove advantageous. There are specific limitations to the present means of production of electronic music of the types reviewed in Chapter 3, one of which is the time and labor required to prepare the music. This is a tedious business requiring painstaking effort. There is also at the present time the need for expensive and elaborate electronic equipment, such as the RCA Electronic Music Synthesizer, if a wide variety of sounds are to be provided. In view of what we have already said in regard to such equipment, it is obvious that the development of simpler equipment operated via computer programming would serve the composer's needs as well as the music analyst's. This should be an interesting engineering problem in itself, since it involves taking from the digital computer a signal representing some sort of musical pulse and converting this signal to output recorded directly on magnetic tape for playback on tape machines. Obviously, playback and distribution of musical output not intended for live performance would be much simplified. The problem of converting digital-computer output into

sound is a typical example of digital-to-analog conversion, a field of interest attracting considerable attention at the present time. Digital computers as they are now constructed are incapable of "doing anything"; they simply supply answers to problems. On the other hand, the second class of computers, called *analog computers*, simulate by analogy, usually electrical, actual processes which may be mechanical as well as purely mathematical. Electrical circuits have certain properties which can be used to simulate addition, subtraction, multiplication, and division, and even integration and differentiation of mathematical functions. It is in recognition of this property of electrical circuits that we can build analog computers which operate upon continuous functions rather than discrete integers as do digital computers. Details of the principles of operation of analog computers can be found in standard reference works.¹¹ Digital-to-analog conversion is required in the present instance, since the output desired is recorded sound. The study of this problem could be initiated in several ways. For example, computer output might possibly be used to activate signals from sine-wave generators, which in turn would be fed to the recording head of a magnetic tape recorder. A second procedure would be to adapt one of the digital-to-analog converters now on the market, such as the digital-to-voltage converter manufactured by Epsco, Inc., Boston, Massachusetts. The combination: *digital computer, digital-to-voltage converter, and magnetic tape recorder* seems potentially the most satisfactory composing instrument in terms of the technology of today. In such a device, sound synthesis would be carried out mathematically in the computer. Tentatively, this seems to be a technique superior to that of activating devices such as the RCA Music Synthesizer by means of coded taped input prepared by computer programming.

A related problem is the notation for various types of synthetic music. As mentioned in Chapter 3, a notation has been devised for the *elektronische musik* produced at Cologne, but it also seems apparent that a more general notation would be desirable. This notation might, for example, be based upon the analysis of sounds in terms of digital information. The processing of this information in a computer by the composer to produce a written score as well as recorded sound should be highly efficient compared with current methods of composition. Yet another related appli-

¹¹ For example, I. A. Greenwood, Jr., J. V. Holdam, Jr., D. Macrae, Jr., *Electronic Instruments*, vol. 17 of the MIT Radiation Laboratory Series, McGraw-Hill Book Company, Inc., New York, 1958; G. A. Korn and T. M. Korn, *Electronic Analog Computers*, 2d ed., McGraw-Hill Book Company, Inc., New York, 1956. Also periodicals such as *Instruments and Automation*.

cation of considerable interest in this combined area of computer and electronic music would be the realization of music too difficult or too complex for performance by live performers. Not an inconsiderable amount of modern music verges on being almost too difficult to perform. Charles Ives' *Fourth Symphony*, for example, is a case in point, to say nothing of a large body of more recent works. It would seem that the threshold level of our ability to perceive complex rhythms and tonal combinations exceeds present performance capacities.

6. It should be mentioned, although not directly as a consequence of the present computer experiments, that certain mechanical aids to the composer might be effected by means of a computer. In particular, one great help would be the copying of parts from scores to eliminate what is now very tedious and time-consuming work. This could be readily accomplished once the scanning devices referred to earlier come into use. Moreover, these scanning devices could be used to read composer's written scores and convert them into sound as well as into printed parts.

7. It is also necessary to take note of one less attractive possibility, but one which must also at least be mentioned, since it is so often suggested. This is the efficient production of banal commercial music. For example, it is not difficult to conceive of programs for writing music of this sort to generate songs for juke-box consumption and similar uses, probably at a highly efficient and rapid rate. All applications of this sort, however, are nonartistic and fall outside the area of problems of aesthetic interest. Belonging in a somewhat similar category is the frequently asked question of whether synthetic Beethoven, Bartók, or Bach might also be produced by computers. The answer to this would seem to depend on the degree to which the elements of the styles of these composers could be verbalized, i.e., coded in a form suitable for computer programming. Appropriate statistical analysis of Beethoven's music might conceivably lead to the production of synthetic average Beethoven, just as, in a sense, the application of strict counterpoint rules can yield a reasonable simulation of average sixteenth-century style, *quite independently of whether computers or normal composing techniques are employed*. The goal rather than the means appears objectionable here, however. The conscious imitation of other composers, by any means, novel or otherwise, is not a particularly stimulating artistic mission. Moreover, this type of study is, in the final analysis, a logical tautology, since it produces no information not present initially. The statistical aspect of the problem should not be permitted to obscure this point. Reduced to its extreme case, this process would revert

to coding exactly the content of a specific and particular piece of music, feeding this information into a computer, and obtaining back from the computer exactly the same piece of music. It is obvious that nothing is accomplished by such an operation.

8. The preparation of computer music for other forms of communication such as the stage, films, and broadcasting might be of particular interest in the use of computer music in the combinations with synthetic music discussed later on. As noted in Chapter 3, financial support in Europe for experiments in electronic music has been provided by government radio stations. This support is given because of the possible exploitation of this other species of experimental music in broadcasting and related forms of communication. There is no reason to suppose that effective means of producing musical and sound continuity for this type of end use might not be carried out efficiently by one or another of the various means we have suggested.

Summary of Results

If we consider the various results included in the *Illiac Suite*, it seems reasonable to conclude that the basic objectives initially outlined for this project have been achieved. It has been shown that computers provide a novel means for studying and exploiting certain techniques of musical composition which can be utilized to produce both conventional and unconventional musical structures. Secondly, by applying new concepts it has been possible to illustrate how various musical forms and processes of composition can be treated from a new viewpoint with the aid of computers. In concluding this discussion, it is desirable to summarize what seem to us the more significant results of these experiments.

1. The elements of musical communication have been separated into those subject to analysis, to coding, and to transcription, namely, specific elements of actual musical structures, and into those more intangible elements which, in general, appear to elude highly specific characterization. It is only the first group of musical elements which were considered subject to coding and to experimentation with computers at the present time.

2. The relationship of information theory to musical problems was discussed to point out how this theory is useful in setting up a technique for musical composition with a digital computer. The specific technique of restricted random number processes was selected as that most desirable for the simulation of composing procedures.

3. The process of musical composition was then defined in terms of the extracting of orderly structures out of random materials by a process of selection and rejection. Specifically, the process of musical composition was regarded as a process of introducing redundancy into a random musical situation.

4. A technique was then devised for studying the processes of musical composition by means of an automatic high-speed digital computer such as the Illiac, the computer located at the University of Illinois. The technique depends upon the Monte Carlo method for generating random numbers. The notes of the scale are numbered in sequence upwards, and then the computer is used to generate random numbers which can be interpreted as the notes of random music. The effectiveness of musical rules in bringing order into this musical texture can then be studied by operating upon these numbers with mathematical processes which express rules of musical composition.

5. It has been shown that historically well-known compositional techniques, problems of current interest to composers, and even more novel processes of composition can all be investigated by this approach. In particular, four Experiments were completed in which the following four sets of problems were successively examined: (a) The writing of simple diatonic melodies and the writing of simple two-part four-part diatonic polyphony. (b) The writing of four-part first-species counterpoint. This problem was studied to provide an example of how a well-known compositional technique can be adapted to computer processing. (c) A study of the programming of rhythm, dynamics, playing instructions, random and simple chromatic music, and of interval and tone rows was carried out. This was done to demonstrate how a computer might be applied to modern compositional problems. (d) A more abstract formulation for musical composition, that is, the application of certain techniques of probability theory and, more particularly, of information theory to produce a number of examples of Markoff chain music, was examined. This last project was carried out to initiate a study of whether a more fundamental basis than the conventional rules of composition might exist for imparting order to musical structures.

6. Enough musical output was produced by the computer to permit the assembly of a presentation of the experimental results in the form of a four-movement composition for string quartet, which we have entitled the *Illiac Suite for String Quartet*.

7. Lastly, some suggestions, mostly based upon the results contained in

the *Illiac Suite*, have been presented to indicate how the present work might be extended both in terms of immediate tasks and in terms of longer-range, more speculative projects. Specific examples of applications in the fields of music analysis and music composition were proposed.

In retrospect, it is interesting to compare the techniques applied to produce music by means of the Illiac with some of the comments of Stravinsky in regard to musical composition quoted in Chapter 3. When Stravinsky spoke of art as "the contrary of chaos," and said that "we feel the necessity to bring order out of chaos," that "tonal elements become music only by virtue of their being organized," and that we must "proceed by elimination—to know how to *discard*," he was speaking, of course, primarily in general terms rather than in terms of detailed specific method. However, it is pleasant to note how accurately these ideas do indeed seem to express the logic inherent in the process of musical composition and how, perhaps for the first time, some considerable measure of quantitative significance can be attached to aesthetic commentary. The net result in the long run can only lead toward a clarification of aesthetic criteria, the improvement of musical understanding, and the production of new types of musical compositions.

ILLIAC SUITE FOR STRING QUARTET

I. EXPERIMENT NO. I

L.A.HILLER, JR. AND L.M.ISAACSON

PRESTO

(A)

(B)

90

(C)

110

120

130

140

150 ANDANTE

160

(E)

Musical score for section (E) starting at measure 189. The score consists of eight staves for different instruments. Measure 189 begins with a dynamic ff. Measures 190-191 show various rhythmic patterns with dynamics f and ff. Measure 192 starts with a dynamic ff.

193

Continuation of the musical score for section (E) from measure 193. The score shows a continuation of the rhythmic patterns and dynamics from the previous measures, with a dynamic ff appearing in measure 393.

200 ALLEGRO

Continuation of the musical score for section (E) from measure 200. The score includes dynamics such as ff, pp, and ff, along with various rhythmic patterns. The section concludes with a dynamic ff.

210

Continuation of the musical score for section (E) from measure 210. The score maintains a fast tempo (Allegro) with various rhythmic patterns and dynamics, including ff and ff.

220

Continuation of the musical score for section (E) from measure 220. The score continues with a fast tempo and various rhythmic patterns, featuring dynamics such as ff, ff, pp, and ff.

150 (F) 152

150 (F) 152

153

153

155

155

270

270

II. EXPERIMENT NO.2

(A) ADAGIO, MA NON TROPPO LENTO



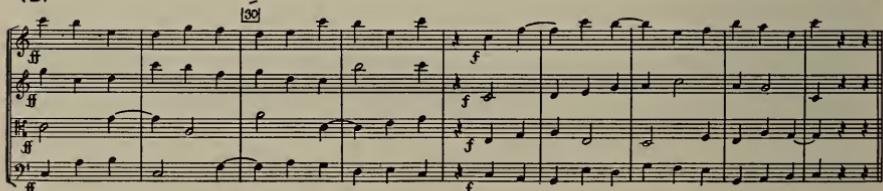
(B)



(C)



(D)



(E)

40

pp
pp
pp
pp

(F)

50

ff
ff
ff
ff

(G)

60

ff
ff
ff
ff

(H)

66

p
p
p
p

CODA

70

ff
ff
ff
ff

III. EXPERIMENT NO. 3

ALLEGRO CON BRIO

THIS SYMBOL

M E A N S S T R I K E T H E B O D Y O F T H E I N S T R U M E N T W I T H K N U C K L E S .

<img alt="Musical score page 5 showing measures 50-60. The score continues with four staves. Measures 50-51 show 'pp' and 'cresc.' markings. Measures 52-53 show 'pp' and 'cresc.' markings. Measures 54-55 show 'pp' and 'cresc.' markings. Measures 56-57 show 'pp' and 'cresc.' markings. Measures 58-59 show 'pp' and 'cresc.' markings. Measures 60-61 show 'pp' and 'cresc.' markings. Measures 62-63 show 'pp' and 'cresc.' markings. Measures 64-65 show 'pp' and 'cresc.' markings. Measures 66-67 show 'pp' and 'cresc.' markings. Measures 68-69 show 'pp' and 'cresc.' markings. Measures 70-71 show 'pp' and 'cresc.' markings. Measures 72-73 show 'pp' and 'cresc.' markings. Measures 74-75 show 'pp' and 'cresc.' markings. Measures 76-77 show 'pp' and 'cresc.' markings. Measures 78-79 show 'pp' and 'cresc.' markings. Measures 80-81 show 'pp' and 'cresc.' markings. Measures 82-83 show 'pp' and 'cresc.' markings. Measures 84-85 show 'pp' and 'cresc.' markings. Measures 86-87 show 'pp' and 'cresc.' markings. Measures 88-89 show 'pp' and 'cresc.' markings. Measures 90-91 show 'pp' and 'cresc.' markings. Measures 92-93 show 'pp' and 'cresc.' markings. Measures 94-95 show 'pp' and 'cresc.' markings. Measures 96-97 show 'pp' and 'cresc.' markings. Measures 98-99 show 'pp' and 'cresc.' markings. Measures 100-101 show 'pp' and 'cresc.' markings. Measures 102-103 show 'pp' and 'cresc.' markings. Measures 104-105 show 'pp' and 'cresc.' markings. Measures 106-107 show 'pp' and 'cresc.' markings. Measures 108-109 show 'pp' and 'cresc.' markings. Measures 110-111 show 'pp' and 'cresc.' markings. Measures 112-113 show 'pp' and 'cresc.' markings. Measures 114-115 show 'pp' and 'cresc.' markings. Measures 116-117 show 'pp' and 'cresc.' markings. Measures 118-119 show 'pp' and 'cresc.' markings. Measures 120-121 show 'pp' and 'cresc.' markings. Measures 122-123 show 'pp' and 'cresc.' markings. Measures 124-125 show 'pp' and 'cresc.' markings. Measures 126-127 show 'pp' and 'cresc.' markings. Measures 128-129 show 'pp' and 'cresc.' markings. Measures 130-131 show 'pp' and 'cresc.' markings. Measures 132-133 show 'pp' and 'cresc.' markings. Measures 134-135 show 'pp' and 'cresc.' markings. Measures 136-137 show 'pp' and 'cresc.' markings. Measures 138-139 show 'pp' and 'cresc.' markings. Measures 140-141 show 'pp' and 'cresc.' markings. Measures 142-143 show 'pp' and 'cresc.' markings. Measures 144-145 show 'pp' and 'cresc.' markings. Measures 146-147 show 'pp' and 'cresc.' markings. Measures 148-149 show 'pp' and 'cresc.' markings. Measures 150-151 show 'pp' and 'cresc.' markings. Measures 152-153 show 'pp' and 'cresc.' markings. Measures 154-155 show 'pp' and 'cresc.' markings. Measures 156-157 show 'pp' and 'cresc.' markings. Measures 158-159 show 'pp' and 'cresc.' markings. Measures 160-161 show 'pp' and 'cresc.' markings. Measures 162-163 show 'pp' and 'cresc.' markings. Measures 164-165 show 'pp' and 'cresc.' markings. Measures 166-167 show 'pp' and 'cresc.' markings. Measures 168-169 show 'pp' and 'cresc.' markings. Measures 170-171 show 'pp' and 'cresc.' markings. Measures 172-173 show 'pp' and 'cresc.' markings. Measures 174-175 show 'pp' and 'cresc.' markings. Measures 176-177 show 'pp' and 'cresc.' markings. Measures 178-179 show 'pp' and 'cresc.' markings. Measures 180-181 show 'pp' and 'cresc.' markings. Measures 182-183 show 'pp' and 'cresc.' markings. Measures 184-185 show 'pp' and 'cresc.' markings. Measures 186-187 show 'pp' and 'cresc.' markings. Measures 188

A musical score page showing ten measures of music for orchestra and piano. The score includes parts for Flute, Clarinet, Bassoon, Trombone, and Piano. The piano part features a basso continuo line. The music is in common time. Measure 1 starts with a forte dynamic (f) and includes dynamics such as f, ff, pp, cresc., mp, pizz., and ff col legno. Measure 2 contains a melodic line for flute and clarinet. Measures 3-4 show a bassoon solo. Measures 5-6 feature a bassoon and piano duet. Measures 7-8 show a bassoon and piano duet. Measures 9-10 show a bassoon and piano duet.

(C)

A page from a musical score for orchestra, specifically the Adagio section. The page is numbered 80 in the top right corner. The music is written in 2/4 time with a key signature of one sharp. The score includes parts for strings (Violin I, Violin II, Viola, Cello), woodwinds (Oboe, Bassoon), brass (Trombone, Horn), and percussion (Drum). The notation shows various dynamics like forte (f), piano (p), and accents. The vocal part consists of three staves of lyrics in German, with the first two staves in soprano range and the third in alto range.

(D) TEMPO I

* WHOLE-TONE SHAKE

A detailed musical score page showing measures 10 through 14. The score includes parts for strings, woodwinds, brass, and piano. Measure 10 starts with a dynamic of ff. Measure 11 begins with a dynamic of p. Measure 12 features a dynamic of pp. Measure 13 begins with a dynamic of f. Measure 14 ends with a dynamic of ff. Various performance instructions are scattered throughout, such as 'martellato' (staccato), 'cresc.' (crescendo), 'sust. forte' (sustained forte), 'dim.' (diminuendo), and 'sust. forte' (sustained forte). Measure 14 concludes with a dynamic of ff dim.

A detailed musical score page showing ten measures of music. The key signature is B-flat major (two flats). Measure 1 starts with a forte dynamic (f) in the strings. Measures 2-3 show woodwind entries with grace notes and slurs. Measure 4 features a piano solo with eighth-note patterns. Measures 5-6 continue with woodwind and piano parts. Measure 7 includes a dynamic instruction 'pp' (pianissimo). Measure 8 has a 'dum' bass drum instruction. Measures 9-10 conclude with woodwind entries and a final dynamic instruction 'ff' (fortissimo).

(G) ADAGIO

(H) TEMPO I

(I)

(J)

(K)

A detailed musical score page showing measures 220 through 225. The score includes parts for strings, woodwinds, brass, and piano. The piano part is prominent, featuring complex chords and rhythmic patterns. Various dynamics like 'cresc.', 'f', 'ff', and 'p' are indicated. Performance instructions such as 'Rit.' (ritardando), 'pizz.' (pizzicato), 'arco sul pont.', 'arcop.', and 'modo ord.' (normal mode) are scattered throughout the score. Measure 225 concludes with a dynamic 'ff' followed by a fermata.

(K) CODA
ADAGIO

Musical score for orchestra, page 10, measures 250-251. The score consists of five staves. The first three staves are labeled "ALLEGRO CON BRIO" and the last two are labeled "ADAGIO". Measure 250 starts with a dynamic of pp and ends with ff . Measure 251 starts with a dynamic of ff and ends with pp .

IV EXPERIMENT NO. 4

TANTO PRESTO CHE POSSIBILE

160

pp cresc poco a poco (al ff)

165

170

175

180

185

190

195

200

(A)

pp cresc. poco a poco (al ff)

(A)

40

41

42

43

44

(B)

(C)

51

52

53

54

55

56

(D)

(E)

90

Bva.

100

(F)

110

(G)

120



(H) (I)

(H) (J) (I)

(K)

CODA

(L)

160

170

180