# CHAPTER 6

## CONDITIONAL SELECTION I: MARKOV CHAINS

Markov chains were formulated by the Russian mathematician Andrei Andreevich Markov (1856-1922) to model sequences of incidents in which each incident affects its immediate successor. Markov himself illustrated this construct by talleying the number of times each pair of consecutive letters occurs in Aleksandr Pushkin's Eugene Onegin in order to distill tendencies of spelling in written Russian. The dependency of consecutive letters in literary texts should be clear from such basic rules of English spelling as "always follow Q with U" and "I before E except after C". Markov chains can also be used to model random processes such as the changing fortunes of a gambler, the inventory of a stocked commodity stocked under pressure from continuing demand, and genetic fluctuations under random matings and mutations.

Since each incident in a Markov chain corresponds to a change in state between two consecutive points in time, we commonly refer to the incidents themselves as transitions. It is convenient to designate the state of the chain existing prior to a transition as the transition's source (equivalently, the chain's past state) while designating the outcome of the transition as the transition's destination (the chain's current state). Consecutive transitions link up so that the destination for one transition serves as the source for the next. The range of a Markov chain is, obviously, the collection of all possible states. This collection is always discrete (heading 4.1), though it may be either bounded or unbounded. Bounded chains have a finite number of states; for example, the Roman alphabet has 26 letters. Unbounded chains can grow arbitrarily large; for example, the number of bacteria in a culture can be modeled as a Markov chain relating the current number to the number existing one minute previously, and such numbers can increase indefinitely.

## 6.1 EXAMPLE: TWO UNBOUNDED MARKOV CHAINS

Figures 6-1 and 6-2 illustrate two unbounded Markov chains at work. These Figures illustrate both consecutive positions in each chain and motions between positions. The range of positions spans all the integers:

... -4, -3, -2, -1, 0, 1, 2, 3, 4, ...

where 0 indicates the first position in each chain. This range is unbounded in both the positive and negative directions; even though neither chain dips below 0 in Figures 6-1 and 6-2, both are entirely capable of doing so. The essential behavior of either chain is inherent in its motions: the likelihoods of moving in either direction are unaffected by the current position (this condition does not hold for Markov chains in general; consider Markov's own example), while the range of motions is limited to the collection:

-2, -1, 0, 1, 2

Figure 6-1: A first-order Markov chain with zeroth-order motions.

Figure 6-2 A second-order Markov chain with first-order motions.

The chain in Figure 6-1 is <u>first-order</u> with respect to positions because the probabilities associated with each position (current state) vary with the most recent position (past state). It is customary to display first-order transition probabilities

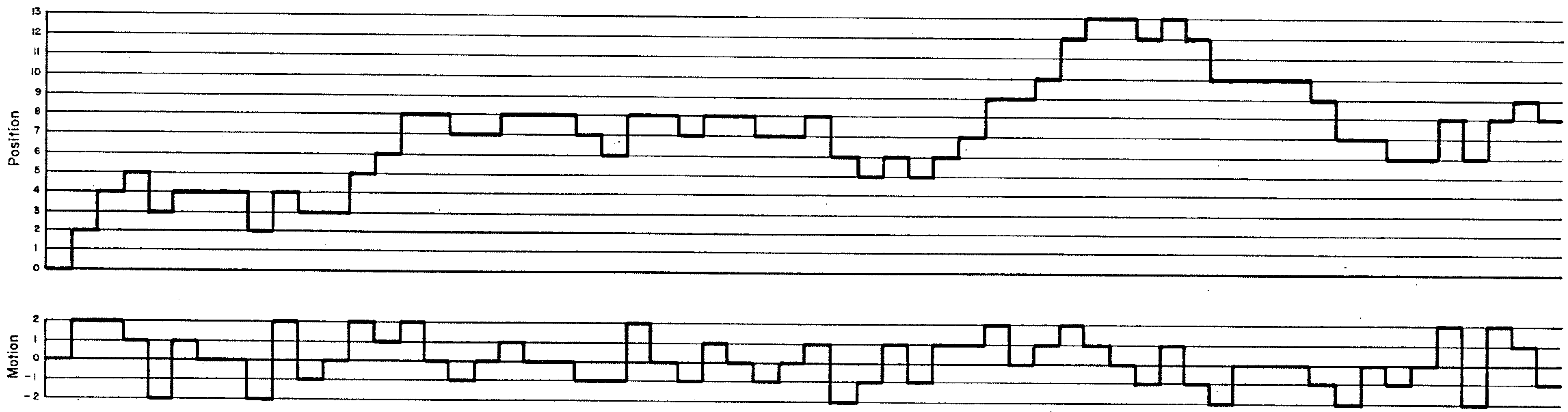

Position
13
12
11
10
9
8
7
6
5
4
3
2
1
0
Motion
2
1
0
-1
-2

Fig 6-1

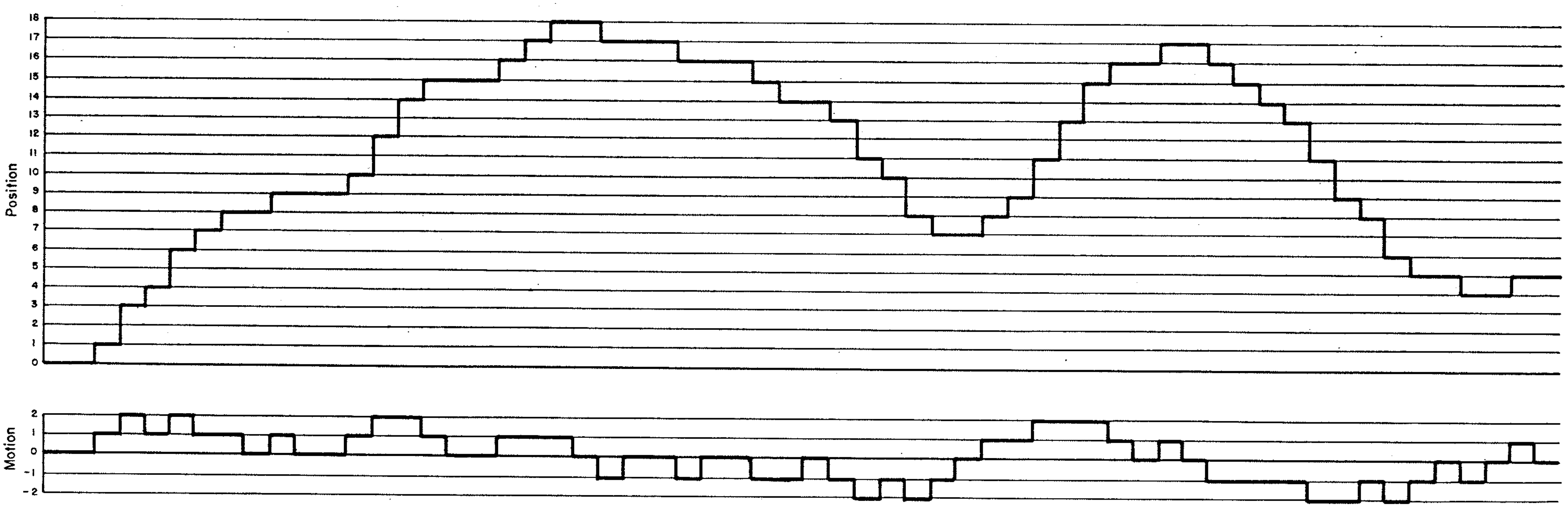

Position
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
Motion
2
1
0
-1
-2

Fig 6-2

| Past Position | Current Position ... | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| : | | : | : | : | : | : | : | : | : | : | |
| -4 | ... | 3/9 | 2/9 | 1/9 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| -3 | ... | 2/9 | 3/9 | 2/9 | 1/9 | 0 | 0 | 0 | 0 | 0 | ... |
| -2 | ... | 1/9 | 2/9 | 3/9 | 2/9 | 1/9 | 0 | 0 | 0 | 0 | ... |
| -1 | ... | 0 | 1/9 | 2/9 | 3/9 | 2/9 | 1/9 | 0 | 0 | 0 | ... |
| 0 | ... | 0 | 0 | 1/9 | 2/9 | 3/9 | 2/9 | 1/9 | 0 | 0 | ... |
| 1 | ... | 0 | 0 | 0 | 1/9 | 2/9 | 3/9 | 2/9 | 1/9 | 0 | ... |
| 2 | ... | 0 | 0 | 0 | 0 | 1/9 | 2/9 | 3/9 | 2/9 | 1/9 | ... |
| 3 | ... | 0 | 0 | 0 | 0 | 0 | 1/9 | 2/9 | 3/9 | 2/9 | ... |
| 4 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 1/9 | 2/9 | 3/9 | ... |
| : | | : | : | : | : | : | : | : | : | : | |

Table 6-1a

| Current Motion | -2 | -1 | 0 | 1 | 2 |
|---|---|---|---|---|---|
| Probability | 1/9 | 2/9 | 3/9 | 2/9 | 1/9 |

Table 6-1b

in square arrays containing one row for each past state and one column for each current state. Such arrays are called Markov matricies (note 1). Since the range of positions in Figure 6-1 is unbounded, the Markov matrix for this chain would have an infinite number of rows and columns. Table 6-1a shows enough elements to communicate the pattern.

Table 6-1a: Position Probabilities in Figure 6-1.

Table 6-1b shows the probabilities associated with each motion in Figure 6-1. With equal probabilities, the motion may be upward, downward, or stationary; motions by one step are twice as likely as motions by two steps. Because the probabilities associated with each motion are unaffected by preceding motions, the motions in Figure 6-1 constitute a "zeroth-order" chain. Zeroth-order Markov chaining is identical with weighted random selection.

Table 6-1b: Motion Probabilities in Figure 6-1.

The chain in Figure 6-2 is second-order with respect to positions because the probabilities associated with each position (current state) vary with the two most recent positions (past state and past-past state). Organizing transition probabilities for a second-order chain, ideally requires a cubic array with a

Past-Past Position: -2

| Past Position | | Current Position -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ... | | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| : | | : | : | : | : | : | : | : | : | : | |
| -4 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| -3 | ... | 1/3 | 1/3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| -2 | ... | 0 | 1/3 | 1/3 | 1/3 | 0 | 0 | 0 | 0 | 0 | ... |
| -1 | ... | 0 | 0 | 0 | 1/3 | 1/3 | 1/3 | 0 | 0 | 0 | ... |
| 0 | ... | 0 | 0 | 0 | 0 | 0 | 2/3 | 1/3 | 0 | 0 | ... |
| 1 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 2 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 3 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 4 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| : | | : | : | : | : | : | : | : | : | : | |

Past-Past Position: -1

| Past Position | | Current Position -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ... | | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| : | | : | : | : | : | : | : | : | : | : | |
| -4 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| -3 | ... | 2/3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| -2 | ... | 1/3 | 1/3 | 1/3 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| -1 | ... | 0 | 0 | 1/3 | 1/3 | 1/3 | 0 | 0 | 0 | 0 | ... |
| 0 | ... | 0 | 0 | 0 | 0 | 1/3 | 1/3 | 1/3 | 0 | 0 | ... |
| 1 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 2/3 | 1/3 | 0 | ... |
| 2 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 3 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 4 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| : | | : | : | : | : | : | : | : | : | : | |

Past-Past Position: 0

| Past Position | | Current Position -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ... | | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| : | | : | : | : | : | : | : | : | : | : | |
| -4 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| -3 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| -2 | ... | 1/3 | 2/3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| -1 | ... | 0 | 1/3 | 1/3 | 1/3 | 0 | 0 | 0 | 0 | 0 | ... |
| 0 | ... | 0 | 0 | 0 | 1/3 | 1/3 | 1/3 | 0 | 0 | 0 | ... |
| 1 | ... | 0 | 0 | 0 | 0 | 0 | 1/3 | 1/3 | 1/3 | 0 | ... |
| 2 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2/3 | 1/3 | ... |
| 3 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 4 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| : | | : | : | : | : | : | : | : | : | : | |

Past-Past Position: 1

| Past Position | | Current Position -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ... | | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| : | | : | : | : | : | : | : | : | : | : | |
| -4 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| -3 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| -2 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| -1 | ... | 0 | 1/3 | 2/3 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 0 | ... | 0 | 0 | 1/3 | 1/3 | 1/3 | 0 | 0 | 0 | 0 | ... |
| 1 | ... | 0 | 0 | 0 | 0 | 1/3 | 1/3 | 1/3 | 0 | 0 | ... |
| 2 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 1/3 | 1/3 | 1/3 | ... |
| 3 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2/3 | ... |
| 4 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| : | | : | : | : | : | : | : | : | : | : | |

Past-Past Position: 2

| Past Position | | Current Position -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ... | | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| : | | : | : | : | : | : | : | : | : | : | |
| -4 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| -3 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| -2 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| -1 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 0 | ... | 0 | 0 | 1/3 | 2/3 | 0 | 0 | 0 | 0 | 0 | ... |
| 1 | ... | 0 | 0 | 0 | 1/3 | 1/3 | 1/3 | 0 | 0 | 0 | ... |
| 2 | ... | 0 | 0 | 0 | 0 | 0 | 1/3 | 1/3 | 1/3 | 0 | ... |
| 3 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1/3 | 1/3 | ... |
| 4 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| : | | : | : | : | : | : | : | : | : | : | |

Table 6-2a

| Past Motion | Curent Motion -2 | -1 | 0 | 1 | 2 |
|---|---|---|---|---|---|
| -2 | 1/3 | 2/3 | 0 | 0 | 0 |
| -1 | 1/3 | 1/3 | 1/3 | 0 | 0 |
| 0 | 0 | 1/3 | 1/3 | 1/3 | 0 |
| +1 | 0 | 0 | 1/3 | 1/3 | 1/3 |
| +2 | 0 | 0 | 0 | 2/3 | 1/3 |

Table 6-2b

| Past State | Current State A | B | C |
|---|---|---|---|
| A | 5/10 | 4/10 | 1/10 |
| B | 4/10 | 5/10 | 1/10 |
| C | 4/10 | 4/10 | 2/10 |

Table 6-3

third dimension to accomodate each different past-past state; on paper, such cubic array may best be depicted using a separate square matrix for each past-past state; the five matricies provided in Table 6-2a illustrate representative cases.

Table 6-2a: Position Probabilites in Figure 6-2.

Motions in Figure 6-2 carry inertia: upward motions tend to propagate further upward motions, while downward motions tend to propagate further downward motions. These motions constitute a first-order chain because the probabilites associated with each motion (current state) vary with the most recent motion (past state). Table 6-2b presents the probabilities for transitions between motions; notice that since the range of motions is finite, the probablities can be all be explicitly organized into a square matrix.

Table 6-2b Motion Probabilites in Figure 6-2.

## 6.2 PROPERTIES OF FIRST-ORDER MARKOV CHAINS

Though chains of second order and higher would seem to provide increasingly generalized formulations of Markov's concept, it can be shown with a little conceptual juggling that any chain of any order can be converted into a first-order representation (note 2). Consequently, most of the mathematical literature on Markov chains deals exclusively with first-order properties.

Two properties which provide useful insights into the behavior of first-order chains are the waiting counts, which reflect how long a chain might linger in an individual state before moving on to the next, and the stationary probabilities, which indicate the proportions in which the various states will occur over the long term. In the discussion which follows, we assume a special kind of chain which is irreducible in the following sense: the chain must be capable of accessing any given state from every other state, either directly or indirectly through two or more transitions. This assumption insures that every state available at the outset remains a viable point of habitation throughout the life of the chain.

### 6.2.1 Waiting Counts

For a first-order Markov chain, we refer to the probability of a state effecting a transition to itself as the waiting probability (or fixed-state probability) and to the number of times a single state occurs consecutively as the waiting count. Waiting counts follow a geometric distribution (heading 4.2.2.3) whose parameter is the waiting probability. If we denote the waiting probability for the ith state as P(i,i), then Equation 6-1 will give the expected waiting count N(i).

$$N(i) = \frac{P(i,i)}{1-P(i,i)} + 1 = \frac{1}{1-P(i,i)} \qquad \text{(Equation 6-1)}$$

As an example, consider the Markov chain whose matrix of transition probabilities is given by Table 6-1a. In this chain, P(i,i) takes the value 1/3 for every position i. The expected waiting count for any position is therefore:

$$N = \frac{1}{1-0.333} = 1.5$$

Notice that the observed average waiting counts for positions in Figure 6-1 deviate around from this expected count: position 6 occurs five times for one unit per occurance, giving an average of 1.0, while position 8 occurs three (complete) times for one unit, three times for two units, and once for three units, giving an average of 1.7.

### 6.2.2 Stationary Probabilites

The relative frequency with which a state occurs during a Markov chain is governed by the stationary (or absolute) probability associated with that state. Stationary probabilities result over the long term from the transition probabilities. We may gain an appreciation of their nature by considering the behavior of many similar chains, all running simultaneously.

Suppose we have 300 Markov chains which share the matrix of transition probabilities (and associated range) given in Table 6-3. Assume that all 300 chains execute their transitions in synchronous and that at the outset, 100 chains reside in state A, 100 in state B, and 100 in state C.

Table 6-3: A three-state Markov matrix.

Consider the first synchronized transition. Of those 100 chains originally residing in state A, we would expect approximately 50 to remain in state A, 40 to jump to state B, and 10 to jump to state C. Similarly, 40 original B's would become A's, 50 would remain B's, and 10 would become C's; while 40 original C's would become A's, 40 would become B's, and 20 would remain C's. The net result of the first transition would therefore be to transform 100 A's, 100 B's, and 100 C's into

| Number of Transitions | Series 1 | | | Series 2 | | |
|---|---|---|---|---|---|---|
| | A | B | C | A | B | C |
| 0 | 100 | 100 | 100 | 300 | 0 | 0 |
| 1 | 130 | 130 | 40 | 150 | 120 | 30 |
| 2 | 133 | 133 | 34 | 135 | 132 | 33 |
| 3 | 133 | 133 | 34 | 134 | 133 | 33 |
| 4 | 133 | 133 | 34 | 133 | 133 | 34 |
| 5 | 133 | 133 | 34 | 133 | 133 | 34 |

Table 6-4

| Average Duration | Waiting Probability | Waiting Count | Average Length |
|---|---|---|---|
| 2 | 0.875 | 8.000 | 16.00 |
| 3 | 0.813 | 5.333 | 16.00 |
| 5 | 0.688 | 3.200 | 16.00 |
| 9 | 0.438 | 1.778 | 16.00 |

Table 6-5

approximately 130 A's, 130 B's, and 40 C's. For an arbitrary transition, if N(A), N(B), and N(C) represent the number of chains residing in states A, B, and C prior to the transition, the numbers after the transition, N'(A), N'(B), and N'(C) may be obtained by the following "system of equations":

```
N'(A)  =  5/10*N(A) + 4/10*N(B) + 4/10*N(C)
N'(B)  =  4/10*N(A) + 5/10*N(B) + 4/10*N(C)
N'(C)  =  1/10*N(A) + 1/10*N(B) + 2/10*N(C)
```

We must apply these equations iteratively to estimate the distribution of states after any number of transitions. Table 6-4 illustrates how the populations of states evolve given two initial configurations: series no. 1 initially assumes 100 A's, 100 B's, and 100 C's, while series no. 2 assumes that all initial chains begin with A. Notice that the proportions in both series quickly settle down to 133/300 (44%) A's, 133/300 (44%) B's and 34/300 (12%) C's. These proportions reveal the stationary probabilities inherent in Table 6-3:

Table 6-4: Evolutions of state-populations according to the transition probabilities given in Table 6-3.

## 6.3 MARKOV CHAINS AND MUSIC

Musical analysts during the late 1940's and the 1950's saw Markov matricies as a means of distilling the norms and deviations of musical styles. Allen Irvine McHose (1947) compiled extensive statistics on contrapuntal practices in Bach's harmonizations of chorales and used these statistics to deduce 'correct' and 'incorrect' practices (note 3). Leonard Meyer was also an early proponent of this approach. In Emotion and Meaning in Music (1956), Meyer cites the "Table of Usual Root Progressions" from Walter Piston's Harmony (1941) as "nothing more than a statement of the system of [conditional] probability which we know as tonal harmony". Meyer's later "Meaning in music and information theory" (1967) attempts to correlate this attitude directly with concepts developed by Claude Shannon.

Shannon's best-known article, "The Mathematical Theory of Communication" (1948) is often cited in connection with Markov processes. This article provides the foundation of what is now more commonly known as information theory. It describes a way of measuring the "information content" of a "message", that is, a sequence of discrete symbols such as a literary text. In Shannon's theory, the least predictable messages have the greatest information content; information decreases as redundancy increases. He models messages as Markov chains, so that when the information content is high, transition probabilities between symbols are close to uniform. Conversely, when redundancy is high, transitions will be strongly biased toward certain patterns of succession.

Once this model has been established, Shannon proceeds to analyze 1) how transmitting such a message, through a "noisy channel" (such as a telegraph wire) results in loss of information, and 2) what safeguards can be imposed to minimize this loss. Attempts to apply Shannon's theory to music generally put the listener's faculties for perceiving musical relationships into the role of the "noisy channel". They then draw inferences concerning how much redundancy may be removed from a musical "message" before the message begins to lose its intelligibility.

### 6.3.1 Illiac Suite, Experiment 4

Experiment 4 of Hiller and Isaacson's Illiac Suite (1957; described 1959) stands as the first direct use of Markov chains to compose music. The Illiac selected consecutive intervals for each instrument according to criteria of harmony (greatest weight to most consonant intervals), proximity (greatest weight to smallest intervals), and combinations of the two. Each instrument proceeds independently. The opening through section D (indicated by rehearsal letters in the score) impose zeroth-order weights, while sections E through G are first-order. Sections H through K apply Markov chains separately to strong and weak beats of the meter.

### 6.3.2 Iannis Xenakis's "Markovian Stochastic Music"

In 1959, Iannis Xenakis wrote three works using Markov chains to control successions of large-scale events: Analogique A for string orchestra, Analogique B for sinusoidal sounds, and Syrmos for 18 strings. Though all three works were manually composed, they each could ~~easily~~ conceivably have been generated by computer.

Xenakis's approach differs strongly from that used by Hiller, et al, in that Xenakis uses procedures similar to those described under heading 6.1.3 to control the mass behavior of many simultaneous chains. The states of his chains are constructs which Xenakis calls "screens". Each screen constitutes a configuration of one or more regions of "musical space" (defined by coordinates of register, dynamics, and rate of activity) in which some number of elementary "grains" of sound may occur (Xenakis, 1971, Chapters II-III).

### 6.3.3 Lejaren Hiller and Robert Baker: Computer Cantata

Hiller and Baker used Markov chains of zeroth through second order in the five Strophes of their Computer Cantata (1963; described 1964). These chains imposed transition probabilities derived from the "Putnam's Camp" movement of Charles Ives's Three Places in New England upon each of the following musical attributes: pitches, durations, dynamics, notes versus rests, and playing styles. Figure 6-3 compares the opening flute passages from each Strophe; similar procedures were employed independently to generate each of the remaining instrumental parts. In describing their work, Hiller and Baker cite the strong influence of Claude Shannon (1948). They are most directly concerned with those aspects of Shannon's theory dealing with "information content", which they attempt to treat as a large-scale musical attribute analogous to tempo, key, thickness of texture, and so on.

Figure 6-3: Comparison of opening flute passages from Strophes I-V of the Computer Cantata - Copyright 1963


STROPHE I (zeroth order)
Fl.
STROPHE II (first order)
Fl.
STROPHE III (second order)
Fl.
STROPHE IV (first order)
Fl.
STROPHE V (zeroth order)
Fl.

Fig 6-3

6-13a

Theodore Presser.

### 6.3.4 Curtis Roads's PROCESS/ING Program

A highly original approach to automated composition described by Curtis Roads (1976), implements Markov chains of extremely high order. Roads has developed a program called PROCESS/ING and used it to create several compositions for tape alone including <u>prototype</u> (1975; described 1975) and <u>Plex</u> (1975, revised 1982). The program is extremely complex and can only be summarized here. It is based around 26 "finite automata", each controlling one of 26 attributes characterizing a cloud of sonic "grains". These attributes include begin time, duration, center frequency, temporal density and registral proximity of grains, and so on. The automata are connected by a 26x26 "interconnection matrix", so that, in general, the attribute specified by an automaton during the n+1st cloud depends both upon the past history of the automaton itself and upon the attributes specified by the remaining automata during the nth cloud. The process as a whole may therefore be regarded as a Markov chain in which the number of possible states is given by the number of ways in which attributes may be combined while the order depends upon how much past history is taken into account by the automata.

### 6.3.5 Other Applications

Applications of Markov processes have been described in articles by Kevin Jones (1981) and Laurie Spiegel (1982). In Jones, the range of states consists of a repertory of musical fragments; how these fragments succeed one another is controlled indirectly by the composer through transition probabilities. Spiegel's "harmonic algorithm" details a 'logic' governing progressions of chords derived from the major scale (I, IV, V, ii, vi, iii).

Petr Kotik has developed an interactive editor for composing with Markov chains. This utility ~~allowed~~ assists Kotik in to creating and modifying tables of transition probabilities, which he has used to compose his <u>Solos and Incidental Harmonies</u> (1983). A unique feature of Kotik's approach was that he often specifies transition probabilities for several <u>versions</u> of the same state. For example, he often describes one set of probabilities for an ascending scale and another set of probabilities for a descending scale; the ascending scale would favor upward motion while the descending scale favored downward motion. Each scale degree therefore receives two entries in the table. Incorporated into the transition probabilities are 'background' probabilities of switching direction.

## 6.4 IMPLEMENTATION

The basis of all algorithms for conditional decision-making is feedback. In the case of first-order Markov processes, it is necessary to feed the most recent choice back into the current decision. For example, ~~one~~ a programmer would implement the first-order chain illustrated in Figure 6-1 by feeding back positions as in line 11 of program CHAIN1.

-- Programming example 6-1: program CHAIN1

The library subroutine MARKOV simulates first-order Markov transitions with bounded ranges. It requires five arguments:

1. RESULT - MARKOV selects a new state and returns an associated value in this location (line 15). RESULT may

Ex 6-1

```
      program CHAIN1
      integer VALUE(5),PLACE,MOTION
      real    WEIGHT(5)
      data VALUE/-2,-1,0,1,2/
      data WEIGHT/0.111,0.222,0.334,0.222,0.111/
C
      PLACE = 0
      do (100 times)
        print *, PLACE
        call SELECT(MOTION,VALUE,WEIGHT,5)
        PLACE = PLACE + MOTION
      repeat
      stop
      end
```

6-16a

be either integer or real.

2. VALUE - Array of values for each state. VALUE must be an array of dimension NUM whose type is identical to RESULT.

3. TRANS - Array of transition probabilities. TRANS must be a real array of dimension NUM by NUM; Each row of TRANS must sum to unity.

4. IDX - Index to most recent state. IDX must be an integer. This argument provides the element of feedback; MARKOV automatically updates IDX to the new state with each call. (Notice that IDX serves as index for the loop in lines 8-13).

5. NUM - Number of states. NUM must be an integer.

While the calling program will typically treat TRANS as a two-dimensional array, MARKOV treats TRANS as one-dimensional. It calculates the location in TRANS of the first element in the IDXth row explicitly from IDX and NUM (line 4) so that the subroutine may handle arrays of different sizes.

Ex 6.2

```
      subroutine MARKOV(RESULT,VALUE,TRANS,IDX,NUM)
      dimension VALUE(1),TRANS(1)
C     Determine first location of first row
      IPTR = NUM * (IDX-1)
C     Generate uniform random number between 0 and 1
      R = RANF()
C     Locate corresponding entry in this row of TRANS
      do (IDX=1,NUM)
        IPTR = IPTR + 1
        T = TRANS(IPTR)
        if (R.le.T) exit
        R = R - T
      repeat
      if (IDX.gt.NUM) stop 'Bad transition densities for MARKOV.'
      RESULT = VALUE(IDX)
      return
      end
```

-- Programming example 6-2: subroutine MARKOV --

In general the order of a Markov chain gives the number of items which must be "fed back" into a decision. Implementing Markov chains of arbitrary order requires a special data structure called a queue (heading 10.2.1), and so is beyond the scope of this chapter. The second-order chain depicted in Figure 6-2 is exceptional in that motions are independent of position. Program CHAIN2 illustrates how this chain would be implemented. It treats motions as a first-order chain (line 15) while feeding back the positions as in CHAIN1 (line 16).

-- Programming example 6-3: program CHAIN2 --

## 6.5 DEMONSTRATION 4: MARKOV CHAINING

Demonstration 4 illustrates how Markov chains might be employed to compose a piece of music. This piece may be regarded as a chain of rhythmic units in the sense of "unit" used to discuss Demonstrations 2 and 3. The structure of Demonstration 4 is simpler than in previous demonstrations because there are no concerted phrases: each of the four attributes characterizing a unit -- duration, articulation, register, and degree -- proceeds independently of the others.

### 6.5.1 Compositional Directives

Control over the musical behavior of Demonstration 4 is exerted through matricies of transition probabilities supplied for each of the four musical attributes listed above. Specific elements of these matricies may be acertained by consulting the listing of program DEMO4: Array TRNAVG (lines 17-20) controls average durations; array TRNART (lines 12-15) controls articulations; array TRNREG (lines 34-40) controls registers; finally, array TRNTVL (lines 22-32) controls intervals and -- indirectly -- chromatic degrees.

All long-term evolutions which occur in this piece arise solely as consequences of tendencies inherent in these transition matricies. In particular, the <u>rate</u> at which these evolutions proceed directly results from the waiting probabilities associated with the individual states: since waiting probabilities for articulations are high, articulations evolve very slowly; by contrast, the waiting probabilites associated with degrees are consistently zero, so no two consecutive notes

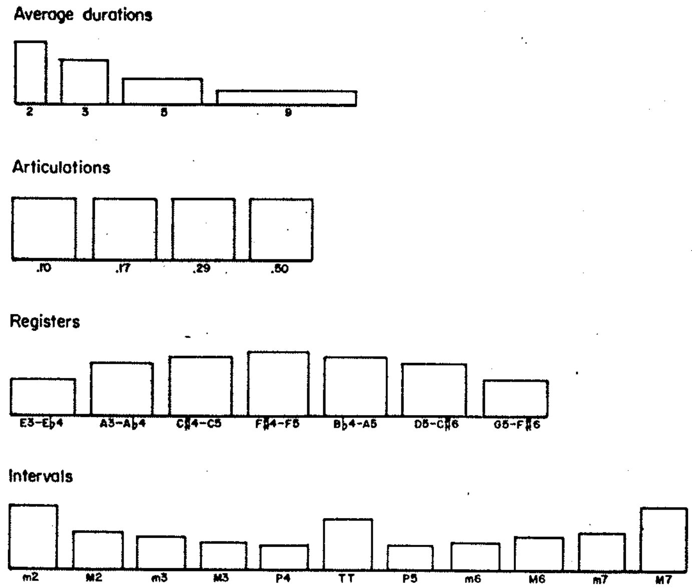

Average durations
2
3
5
9
Articulations
.10
.17
.29
.50
Registers
E3-E♭4
A3-A♭4
C♯4-C5
F♯4-F5
B♭4-A5
D5-C♯6
G5-F♯6
Intervals
m2
M2
m3
M3
P4
TT
P5
m6
M6
m7
M7

Fig 6-4

repeat the same degree. Also pertinent to the long-term behavior are the stationary probabilities depicted in Figure 6-4. Each graph in Figure 6-4 was derived from the corresponding transition matrix using the procedures described under heading 6.2.2.

Figure 6-4: Stationary probabilities for Demonstration 4.

6.5.1.1 Duration - There are four average durations available to any note. The waiting probabilities for all four values have been calculated so that the chain lingers on each value for approximately two measures before moving on to a new value. Table 6-5 illustrates how the average length over which a value holds sway may be computed as the product of the average duration itself and the waiting count. We should recognize that these controls over how long an average duration holds sway are very loose since both the waiting count and the individual durations are random (note 5).

Table 6-5: Expected lengths for sequences of notes with fixed average durations in Demonstration 4.

transition probabilities emphasize the smaller shifts in register.

6.5.1.4 Chromatic Degrees - The progression of chromatic degrees may be regarded as a first-order Markov chain of clockwise displacements around the chromatic circle illustrated in Figure 1-2a. Figure 6-5 details the relative weights associated with each pair of consecutive displacements. Though intervals range from 1 to 11 (unisons are excluded), "wrap-around" arithmetic keeps the current degree always within the range from 1 to 12.

Figure 6-5: Stylistic matrix for Demonstration 4 - Each entry depicts a pair of two rising chromatic intervals with the middle degree fixed arbitrarily at B. The number of semitones in an interval corresponds to the extent of clockwise displacemement around the chromatic circle illustrated in Figure 1-2a.

This approach resembles the INTERVAL feature of Gottfried Michael Koenig's PROJECT2 program. Such matricies will be frequently employed in subsequent Demonstrations and will be designated in this book as stylistic matrices (note 4). The

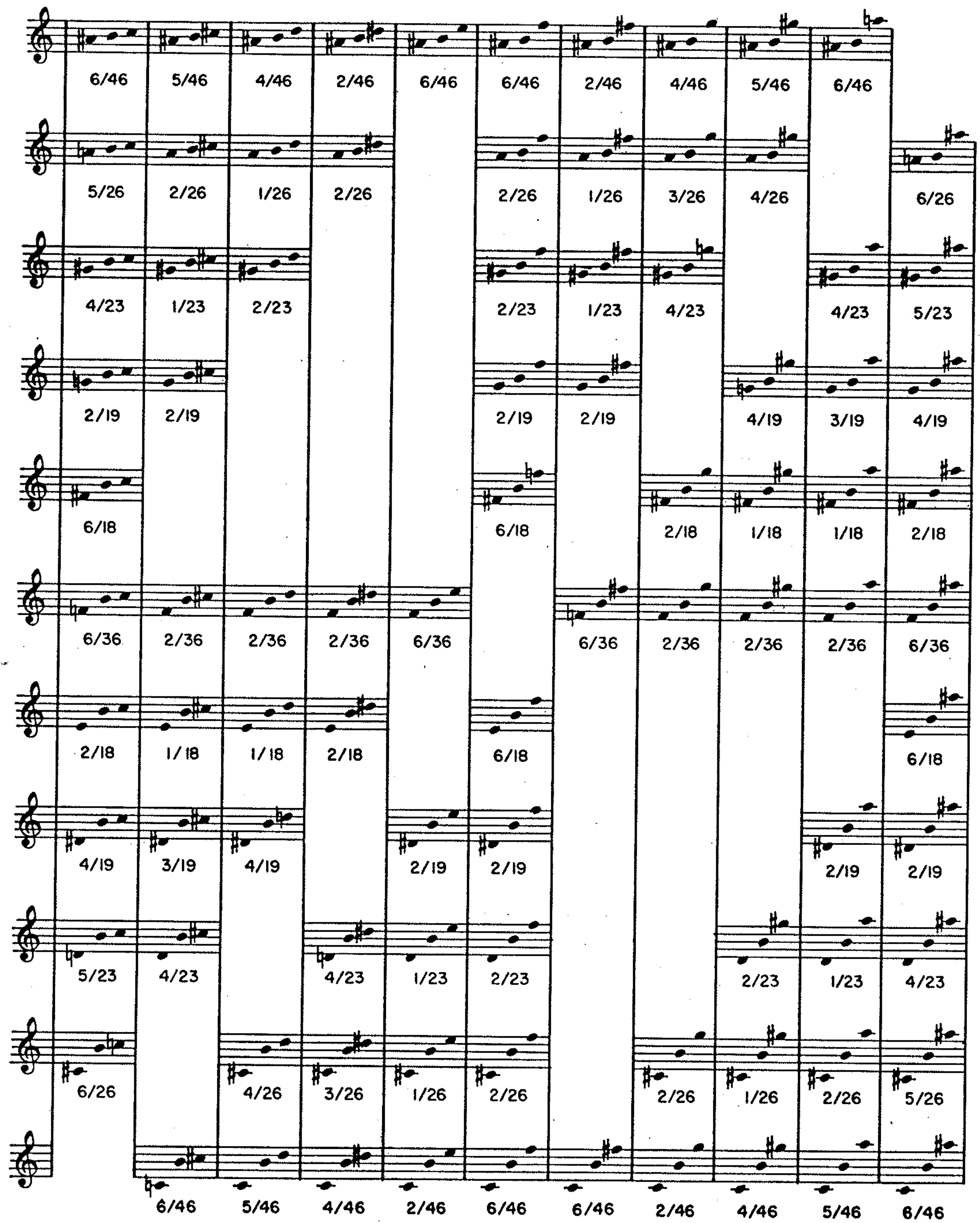

6/46 5/46 4/46 2/46 6/46 6/46 2/46 4/46 5/46 6/46
5/26 2/26 1/26 2/26 2/26 1/26 3/26 4/26 6/26
4/23 1/23 2/23 2/23 1/23 4/23 4/23 5/23
2/19 2/19 2/19 2/19 4/19 3/19 4/19
6/18 6/18 2/18 1/18 1/18 2/18
6/36 2/36 2/36 2/36 6/36 6/36 2/36 2/36 2/36 6/36
2/18 1/18 1/18 2/18 6/18 6/18
4/19 3/19 4/19 2/19 2/19 2/19 2/19
5/23 4/23 4/23 1/23 2/23 2/23 1/23 4/23
6/26 4/26 3/26 1/26 2/26 2/26 1/26 2/26 5/26
6/46 5/46 4/46 2/46 6/46 6/46 2/46 4/46 5/46 6/46

Fig 6-5

current matrix acts both to enforce absolute constraints and to promote desirable tendencies, in this case to encourage a consistently dissonant style. Any triad containing chromatic identities receives a weight of zero, as do major triads, minor triads, augmented triads, and triads with two perfect consonances (for example, C-F-Bb) in any inversion or voicing. The remaining triads receive relative weights from 1 to 6, depending on dissonance. Thus, triads combining a perfect fifth with a minor seventh, (for example, C-G-Bb) always receive a relative weight of 1, regardless of inversion, while triads combining a tritone with a major seventh (for example, C-F#-B) or combining two semitones (for example, C-C#-D) always receive a relative weight of 6.

Figure 6-6 graphs average durations, articulations, and registers in Demonstration 4. The complete product appears in Figure 6-7.

Figure 6-6: Profile of Demonstration 4.

Figure 6-7: Transcription of Demonstration 4.

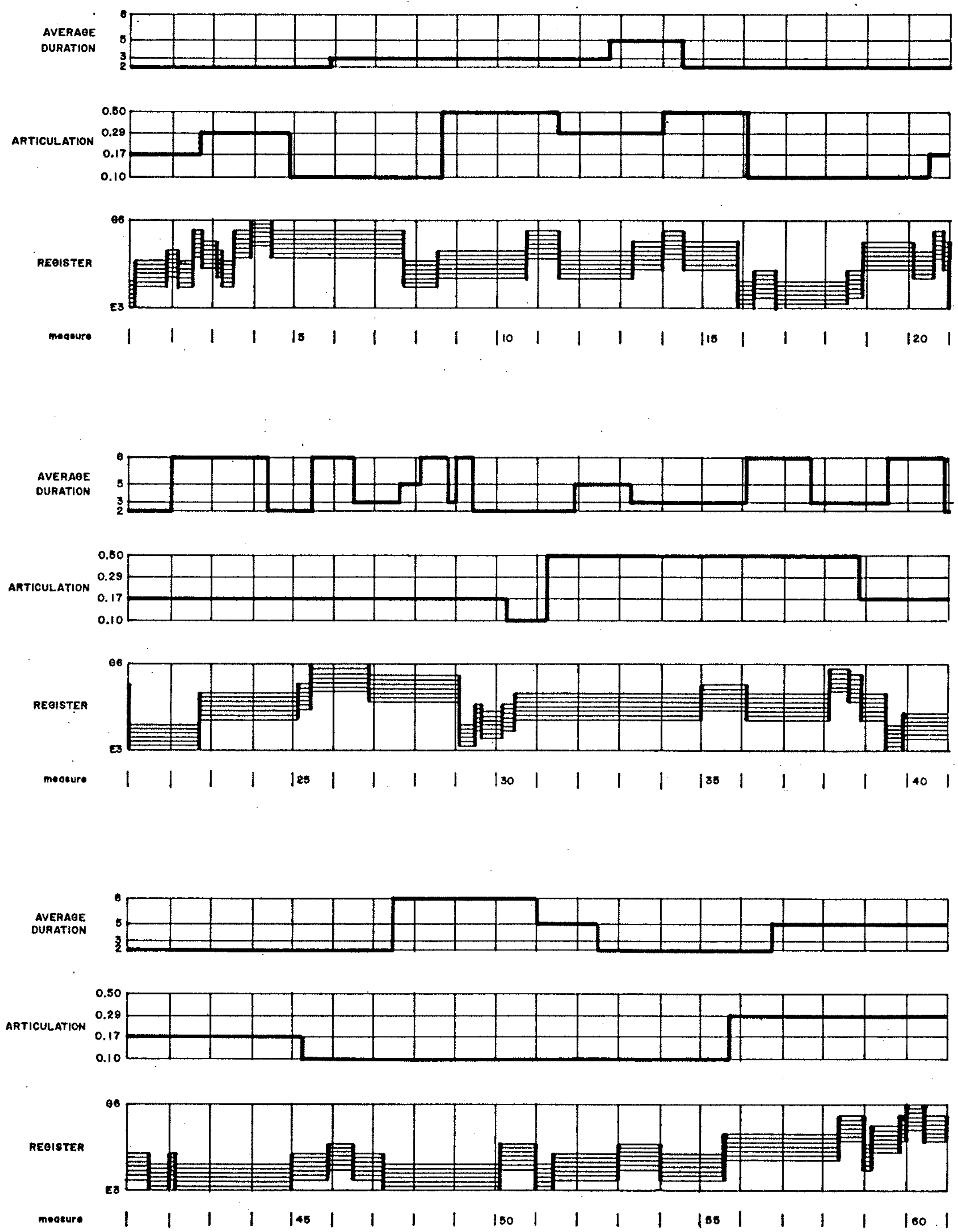

AVERAGE DURATION
6
5
3
2
ARTICULATION
0.50
0.29
0.17
0.10
REGISTER
G6
E3
measure
5
10
15
20
AVERAGE DURATION
ARTICULATION
REGISTER
measure
25
30
35
40
AVERAGE DURATION
ARTICULATION
REGISTER
measure
45
50
55
60

Fig 6-6

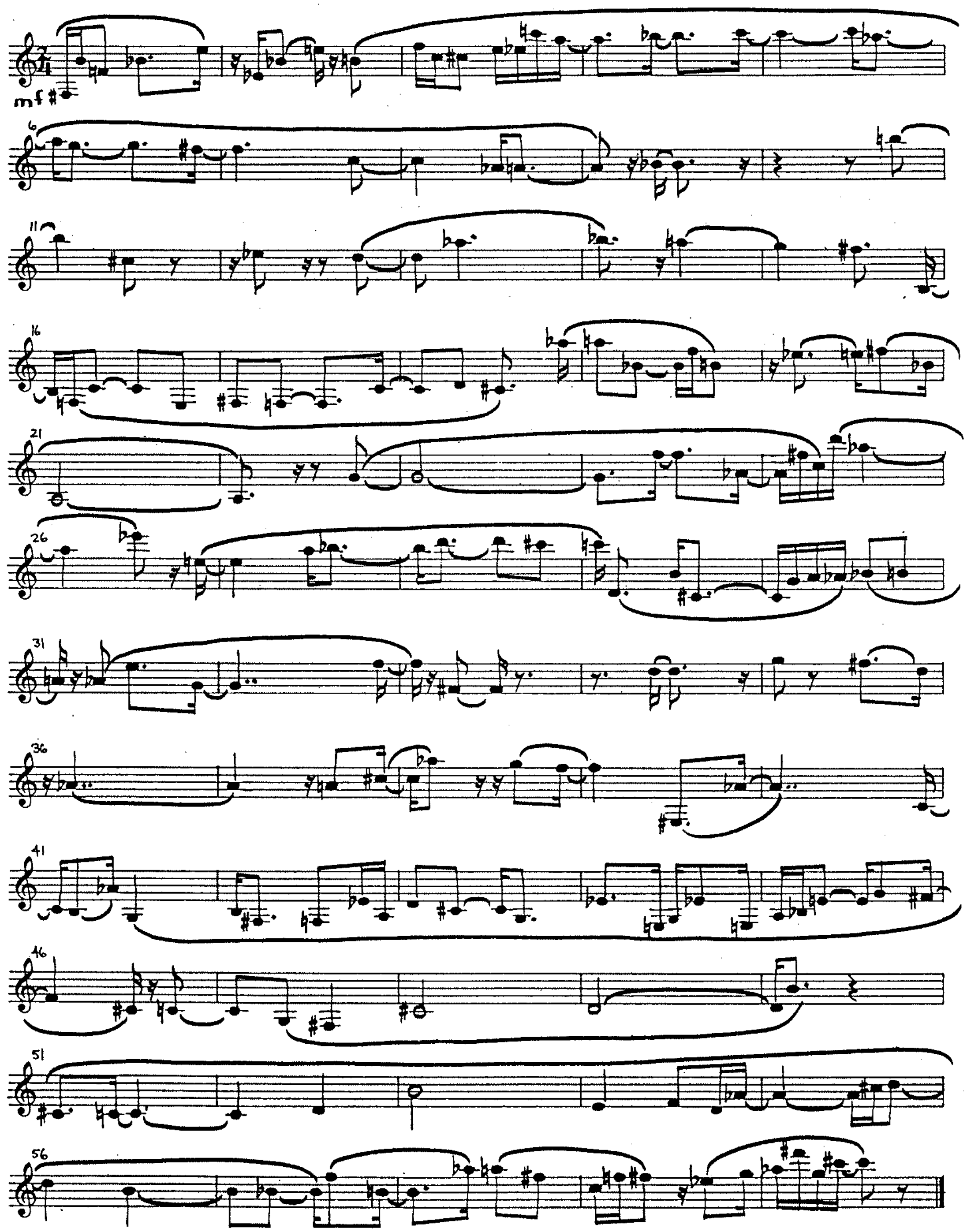
6-236

# Demonstration 4

Clarinet

Charles AMES

STRICTLY ♩ = 80



Ex 6-3

```
      program CHAIN2
      integer VALUE(5),PLACE,MOTION
      real    TRANS(5,5)
      data VALUE/-2,-1,0,1,2/
      data TRANS/0.333,0.667,0.000,0.000,0.000,
     :           0.333,0.334,0.333,0.000,0.000,
     :           0.000,0.333,0.334,0.333,0.000,
     :           0.000,0.000,0.333,0.334,0.333,
     :           0.000,0.000,0.000,0.667,0.333/
C
      PLACE = 0
      IDX = 3
      do (100 times)
        print *, PLACE
        call MARKOV(MOTION,VALUE,TRANS,IDX,5)
        PLACE = PLACE + MOTION
      repeat
      stop
      end
```

6-23c

```
      program DEMO4
c
c     Demonstration of Markov chaining
c
      parameter (MREG=7,MTVL=11,MAVG=4,MART=4)
      integer VALREG(MREG),VALTVL(MTVL)
      real    VALAVG(MAVG),VALART(MART)
      real    TRNREG(MREG,MREG),TRNTVL(MTVL,MTVL),
     :        TRNAVG(MAVG,MAVG),TRNART(MART,MART)
      logical SUCCES,REST
      data VALART/.1,.17,.29,.5/
      data TRNART/.91,.02,.02,.05,
     :            .02,.91,.05,.02,
     :            .05,.02,.91,.02,
     :            .02,.05,.02,.91/
      data VALAVG/2.,3.,5.,9./
      data TRNAVG/.875,.042,.042,.041,
     :            .062,.813,.063,.062,
     :            .104,.104,.688,.104,
     :            .187,.188,.187,.438/
      data VALTVL/1,2,3,4,5,6,7,8,9,10,11/
      data TRNTVL/.13,.11,.09,.04,.13,.13,.04,.09,.11,.13,.00,
     :            .19,.08,.04,.08,.00,.08,.04,.12,.15,.00,.22,
     :            .17,.04,.09,.00,.00,.09,.04,.18,.00,.17,.22,
     :            .11,.10,.00,.00,.00,.10,.11,.00,.21,.16,.21,
     :            .33,.00,.00,.00,.00,.33,.00,.11,.06,.06,.11,
     :            .16,.06,.06,.06,.16,.00,.16,.06,.06,.06,.16,
     :            .11,.06,.06,.11,.00,.33,.00,.00,.00,.00,.33,
     :            .21,.16,.21,.00,.11,.10,.00,.00,.00,.10,.11,
     :            .22,.17,.00,.18,.04,.09,.00,.00,.09,.04,.17,
     :            .22,.00,.15,.12,.04,.08,.00,.08,.04,.08,.19,
     :            .00,.13,.11,.09,.04,.13,.13,.04,.09,.11,.13/
      data VALREG/40,45,49,54,58,62,67/
      data TRNREG/.66,.17,.08,.05,.03,.01,.00,
     :            .11,.66,.11,.06,.03,.02,.01,
     :            .05,.10,.66,.10,.05,.03,.01,
     :            .03,.05,.09,.66,.09,.05,.03,
     :            .01,.03,.05,.10,.66,.10,.05,
     :            .01,.02,.03,.06,.11,.66,.11,
     :            .00,.01,.03,.05,.08,.17,.66/
      data REMAIN/0./, REST/.true./
c
c     Initialize
c
      open (2,file='DEMO4.DAT',status='NEW')
      ITIME = 0
      MTIME = 8 * 60
      IDXART = IRND(MART)
      IDXAVG = IRND(MAVG)
      IDXTVL = IRND(MTVL)
      IDXREG = IRND(MREG)
      IDEG = IRND(12)
c
c     Main composing loop
c
      do
        call MARKOV(ARTIC,VALART,TRNART,IDXART,MART)
        call MARKOV(AVGDUR,VALAVG,TRNAVG,IDXAVG,MAVG)
        if ( REST .or. .not.SUCCES(ARTIC) ) then
c         Select duration of note
          DUR = RANX(AVGDUR,8.0) + REMAIN
          IDUR = max0(1,ifix(DUR+0.5))
          REMAIN = DUR - float(IDUR)
c         Select degree
          call MARKOV(INTRVL,VALTVL,TRNTVL,IDXTVL,MTVL)
          IDEG = IDEG + INTRVL
          if (IDEG.gt.12) IDEG = IDEG - 12
          call MARKOV(IREG,VALREG,TRNREG,IDXREG,MREG)
          REST = .false.
        else
c         Select duration of rest
          DUR = RANX(AVGDUR/2.0,8.0) + REMAIN
          IDUR = ifix(DUR)
          REMAIN = DUR - float(IDUR)
          REST = .true.
        end if
c       Test for end of composition
        if (ITIME+IDUR.gt.MTIME) exit
c       Write note or rest
        call WNOTE(ITIME,IDUR,IDEG,IREG,REST)
      repeat
      close (2)
      stop
      end
```

Ex 6-A

### 6.5.2 Implementation

-- Programming example 6-4: program DEMO4 --

Program DEMO4 implements the one-tiered structure described above as a single composing loop which executes one iteration for each rhythmic unit. All decision-making involving Markov transitions is accomplished through calls to the library subroutine MARKOV (lines 57, 58, 65, and 68).

The symbols of program DEMO4 adhere to four mnemonic 'roots' corresponding to the four musical attributes controlled by the program:

1. AVG - average duration of notes; the average duration of rests is half as large.

2. ART - articulation, expressed as the probability that a rhythmic unit will serve as a rest.

3. REG - register, expressed as the central pitch in a gamut.

4. TVL - chromatic interval, expressed as an integer from 1 to 11.

The number of states available to each attribute is given by a parameter starting with the letter M. Each attribute has an index, required by MARKOV; this index begins with IDX. Values associated with each index reside in arrays beginning with VAL. Arrays beginning with TRN hold transition probabilities for each past and current state.

## 6.6 NOTES

1. The use of the word "matrix" in any context not specifically including a rectangular array or lattice should be avoided as jargon, except perhaps in science fiction novels.

2. Suppose we have a chain of order N with a range of M possible states. Then we can construct a new chain by regarding each possible sequence of N consecutive states in the old chain as a single state of the new chain. This new chain will now have a range of N*M possible sequences. Similar reasoning holds even when the range of possible states is infinite.

3. Though McHose does not mention Markov by name, McHose's approach corresponds exactly to Markov's model.

4. Matrices provide a means of implementing constraints which are highly efficient computationally. They have the advantage of allowing a user to specify preferences on a case-by-case basis and the disadvantage of forcing a user to do so. Not all stylistic traits are as easily formalized into matrices as those depicted in Figure 6-6; constraints against parallel voice-leading for example, are better expressed as rules.

5. The number of random mechanisms affecting a process is often denoted as the <u>degrees of freedom</u>. The more degrees of freedom, the more the individual states of the process deviate from the norm.

## 6.7 RECOMMENDED READING

Cherry, Colin. <u>On Human Communication</u>, 3rd edition (Cambridge: M.I.T. Press, 1978).

Jones, Kevin. "Compositional applications of stochastic processes", <u>Computer Music Journal</u>, volume 5, number 2 (Summer 1981), p. 45.

Hiller, Lejaren and Robert Baker. "Computer Cantata: An investigation of compositional procedure", Perspectives of New Music, volume 3 (Fall-Winter 1964), p. 62