SONOLOGICAL REPORTS

No. 2

THE COMPUTER COMPOSITION - SOUND SYNTHESIS PROGRAMS POD4, POD5 & POD6

Barry D. Truax

1973

INSTITUTE OF SONOLOGY

UTRECHT STATE UNIVERSITY

TABLES AND ILLUSTRATIONS:

TABLE OF CONTENTS

INTRODUCTION

The computer composition - sound synthesis programs POD4, POD5 & POD6 have been designed with the following intentions:

1) To create programs offering as many facilities as possible for real-time sound synthesis within the limited hardware and data storage capability of a PDP-15 computer having as equipment:

- 12 K core storage
- 1 Disk unit
- 2 DECtape units
- 1 real-time clock
- Paper tape reader and punch
- Teletype unit
- D-A converters (8 channels, 12 bits)
- A-D converters (4 channels, 10 bits)
- X-Y plotter

All of these, with the exception of the A-D converters and three channels of the D-A converters are actually used by the programs. The lack of a magnetic digital tape unit precludes mass data storage, and thus the smaller and slower capabilities of the dectape units have been pressed into service for as efficient use as possible.

2) To create programs suitable for use by a composer who directs the flow of activity from the teletype through the various stages of operation, including the making of changes and corrections, the results of which may be heard within a very short period of time. This sense of feedback which is very important for the composer is most often missing in larger computer programs for synthesis where problems of conversion may take days, if not weeks or months, preventing any kind of immediate feed-back of the results of compositional ideas. As well, many of the larger synthesis programs require the user to employ a great amount of psycho-acoustic knowledge in order for him to be able to know what specifications he must make in order to achieve a more or less predictable sound result. It is hoped then that the present programs will not only facilitate this kind of translation of the composer's ideas into sound, but also that they will serve a pedagogical function for those who need first-hand learning experience in working with computer synthesis, and yet have neither the availability of the most advanced synthesis facilities, nor the experience with which to use them fully.

3) To create programs operating from the user's point of view at a higher, syntactical level of composition than that of only the specification of the particulars of sound generation; that is, the composer is not only concerned with the nature of the individual sound event, but also with that of larger groups of events, including entire compositional sections. This kind of compositional thinking, which seems common to the practice of all composers, is seldom if ever given possibility in most computer programs, which often tend to concentrate their resources on the implementation of elaborate features of sound production technique.

4) To create programs that would act as a framework within which the composer could build up a more personal version of the materials available to him, particularly at the level of sound data.

As well, it is hoped that the programs may stimulate development of facilities for sonological research, especially in the development of 'sound comprehending' programs, simulating the process of understanding sound structures as musical structures, and of composition-theoretical programs testing the competence of music users. Although the present programs have been mainly developed and designed for compositional and pedagogical purposes, the course of their development has revealed some further possibilities for progress in this under-developed area.

Of the three programs developed thus far, POD5 & 6 are the most distinct because of the difference in synthesis method, and as well the most suited for compositional work, with POD4 acting as an introduction to them. All use real-time synthesis methods with variable sampling rate, the two major approaches being fixed waveform or 'stored function' synthesis, as in POD4 & 5, and a frequency modulation method in POD6 (first developed by John Chowning at Stanford) producing time-dependent spectra. The POD5 synthesis uses an array of 500, with 60 discrete amplitude levels in the form of scaling factors, and produces sounds in the range of 60 to 8000 Hz. with sampling rates generally between 30 and 47 KHz. Amplitude modulation and two-channel output form additional options in the program. The frequency modulation method is discussed in Section III (p. 6), and in Appendix B, Section III.

For the syntactic aspect discussed in 3) above, it was decided that a statistical distribution represented the most general organization of sound events, and would allow emphasis to be placed on the larger aspects of formal structure, without undue preoccupation with the relationships between individual sound parameters. Given the case of discrete, independent sound events, the Poisson distribution is mathematically the most appropriate (see Appendix A for a fuller account). The basic solution, then, was to organize all events by means of the Poisson distribution, but at the same time give the user the possibility of changing the density of events in time, and restricting the frequency area to "available" and "non-available" regions for each time unit. The sound event itself is described by a frequency and time position determined by the Poisson calculation, and a sound object specification to realize it, meaning waveform, envelope and optionally spatial position and amplitude modulation with POD4 & 5, and envelope, carrier to modulating frequency ratio and maximum modulation index for POD6. The objects are chosen in various statistical ways for each compositional section according to the user's choice. Several performance variables affecting the amplitude-time structure allow the distribution to be realized in diverse ways, and give the user the chance to adjust the output on perceptual grounds.

Because of the restrictions in data storage, simply the parameters needed by the synthesis program are stored in blocks on dectape, with the waveforms (in POD4 & 5) residing on a second dectape, ready to be loaded into core memory when needed. In POD6 a permanent array is calculated by the program and used as the basis for the synthesis process.

In keeping with the fourth intention stated, the data for the basic waveforms can be obtained in various ways within the possibilities of the program itself, for example, by combining segments of waveforms already stored on dectape into new waveforms, or by adding a harmonic structure to one of these existing waveforms. As well, the program accepts waveform data from paper tape and converts it to the appropriate format for the program. By these methods and their combination, the composer can build up an elaborate library of materials with which to work. In POD6, the emphasis shifts away from the waveform itself to the process of frequency modulation and groups or families of complex spectra produced by it - an area where much exploration still needs to be done.

In view of the second intention, the data that must be calculated for each sound event is stored temporarily on dectape, and used by the synthesis part of the program to present the user with the results of a calculated distribution. Changes in any parameters of the calculation, or simply those of the performance characteristics, may then be made and the results assessed. Various kinds of representations of the data being used (numeric, graphic, punched tapes, etc.) are also available to the user to help him make his choices, and facilitate keeping track of changes. Each distribution of events is regarded as a compositional section, and once all changes have been made, the data required to reproduce it with the same performance characteristics is stored on a permanent block of dectape, or within the program storage itself, and a summary is printed on the teletype for the user, giving him the information necessary for recreating the section exactly at a later time. The user can then move on to a new section, whose data is added to that already stored on dectape. At any time, all previous sections, including the one presently being worked on, may be replayed for comparison. For a section of 200 to 250 events, the calculation time prior to loading of waveforms for synthesis is about 20 or 25 seconds. Up to 378 events in a distribution may be heard at any one time before a short interruption is needed to fetch the next block of data.

Many subroutines, including the major synthesis programs, have been kept in as general a form as possible, to allow them to be used in other contexts. Most of the strategies used by the programs are unique to them, and require some familiarization, but some, particularly the types of statistical distributions of waveforms, including the basic tendency mask for the Poisson distribution, have been taken over from their usage in 'Project Two' of G. M. Koenig, of the Institute of Sonology, with very similar functions and specification.

Finally, the author would like to acknowledge the generosity of the Canada Council for financial support, and the Institute of Sonology, Utrecht, for the computer time without which these programs would not have been developed. Also, many thanks must go to G. M. Koenig, C. A. G. M. Tempelaars, and Dr. O. E. Laske, all of the Institute of Sonology for their advice, encouragement and constructive reading of the manuscript during the development stages of the programs.

PREFACE: ON THE MUSICAL IMPLICATIONS OF STATISTICAL PROCEDURES FOR COMPUTER SOUND SYNTHESIS

Basic to all cognitive-perceptual processing of psycho-acoustic information seems to be a sorting and coding process designed to reduce a large mass of input data (such as that involved in sound sequences) to an easily comprehensible sonological representation. If this procedure fails, the input remains unintelligible in that the sonological strategies normally and habitually used by the observer are inadequate to interpret the input. In this case, it is possible that new strategies may be sought for with which a more successful result can be achieved; that is, learning may take place.

In the case of most contemporary music, the organization of the sound material itself (as in electronic music), and/or the syntactical nature of its overall organization (for example, by the methods of total serialism) often results in a surfeit of input information beyond the scope of most observers' perceptual processing capabilities. In this case, it seems that a common sonological strategy reduces the amount of input information regarding localized data, such as short-term relations between near-adjacent sounds, into a kind of statistical appraisal of what is happening, and if possible looks to larger formal or syntactical levels for a recognizable 'shape' which characterizes the input. Suggestive of this process is the experience of expectation aroused in such a situation: one knows <u>not</u> to expect the unexpected. If some overall processing had not been realized, such a non-expectation would be impossible. It seems evident, then, that the reason for much of the emphasis in earlier music on easily recognizable units - melodies, motifs, themes - where the compositional material was organized to support and clarify these units through articulation, was to organize the levels of psycho-acoustic and sonological input so clearly that the observer could interpret it without resorting to the statistical approximation methods described, thereby freeing him to concentrate on musical development and form.

With a complex organization of sounds, however, unless one is trained to perceive them in the manner in which they were composed, the organization of such an artifact will not be readily apparent to the listener, and the result may as well be analyzed statistically and found to obey a more or less random distribution. Thus comes the well-known phenomenon that even explicable events, such as wars, births, etc., may be shown to occur at random from a statistical point of view (i.e. the pattern of their occurrence conforms to a Poisson distribution). Balanced against this is the phenomenon that synthesis based on the results of statistical analysis of a structure does not lead to a reproduction of the structure, or even necessarily to a suitable variant of it, since the musical rules involved in its formation have not themselves been reproduced, but instead only the superficial appearance has been simulated.

Therefore, if one is to become involved with complex sound production and at the same time be concerned with its result being understood as musical, it does not seem inappropriate to use statistical distributions of events or parameters since it can be supposed that most observers are capable of sonological strategies arriving at the recognition of such structures. Note however that if this is done in terms of parameter specification, which is to some extent true of these programs, there is no guarantee that the result will be perceived in the statistical manner of its formation. Central to the competent organization of a distribution is the implicit knowledge of the user as to how the parametric manipulation affects the total sound result; that is, the knowledge of how a theoretical distribution is weighted by perceptual factors. Unfortunately, sonological research is still at such an infant stage that even the basic knowledge regarding such sonic design is obscure, and therefore, since it cannot as yet be incorporated into the present programs, some compensation has to be made by devoting large parts of their operation to the user's testing and perceptual judgment of the sound objects he is working with. With more advanced sound synthesis control, incorporating statistical procedures into the production strategy itself, the sonic categories which still appear in this program as seemingly isolated entities could be unified into more comprehensive units. These would correspond more closely to the reality of the perceptual situation which involves such an interdependence of time-amplitude characteristics that pitch, timbre and loudness, among others, are inextricably intertwined. Therefore, although such further synthesis capability needs to be developed, its intelligent use is contingent on an increased understanding of the nature of mental (i.e. sonological) representation given to sounds, and the strategies developed by music users in forming and manipulating them for purposes of musical problem solving.

Therefore, the sound formation features of the present programs tend to be investigatory in nature, but consequently suitable for learning. On the level of sound sequences, the two most basic concerns for the user are sound densities and statistical organization methods such as tendency masks, since both of these represent the organizational capability of the program in structuring the output in some way analogous to sonological processing. Most general is the concept of density - the rate of sounds in time, which, when combined with the sonological representation of the sound event, gives the rate of information in time. Intelligent use of density control (through the distribution characteristics and performance variables) must be supported by an understanding of the sonological significance levels of the sound materials. Secondly, use of the statistical organization methods, such as:

1) equally weighted probability (alea)
2) variably weighted probability (ratio)
3) variably weighted probability in time (tendency)

represents a simple approach to sonological coding of larger groups of sounds, with the progression from one to the other involving successive simplifications of representation. Presumably, further methods research needs to be done in forming additional possibilities developed from these three basic ones.

The present facility then can be seen as only the most fundamental approach to a truly intelligent program for the production of musical structures, which it is hoped use of the present programs will foster. The value of statistical methods at every stage of such a program seems obvious, but necessarily contingent on the sonological insight incorporated in all methods used to direct the production of a musical result.

I. POISSON DISTRIBUTION OF SOUND EVENTS:

All sound events are distributed by means of a Poisson distribution within a frequency-time area. For a detailed description of the theoretical and practical implications, see Appendix A.

Two further aspects of the distribution are specified by the user. First, the density of events may increase or decrease linearly, depending on the initial density in sounds per second, and the total number of events specified by the user. However, the user may also request an equal theoretical density throughout the distribution, or else specify the distribution characteristics by giving the initial and final densities, leaving the program to calculate the expected number of events. Secondly, a tendency mask specified by the user delineates the available frequency regions throughout the section. Figure 1 illustrates such a mask.

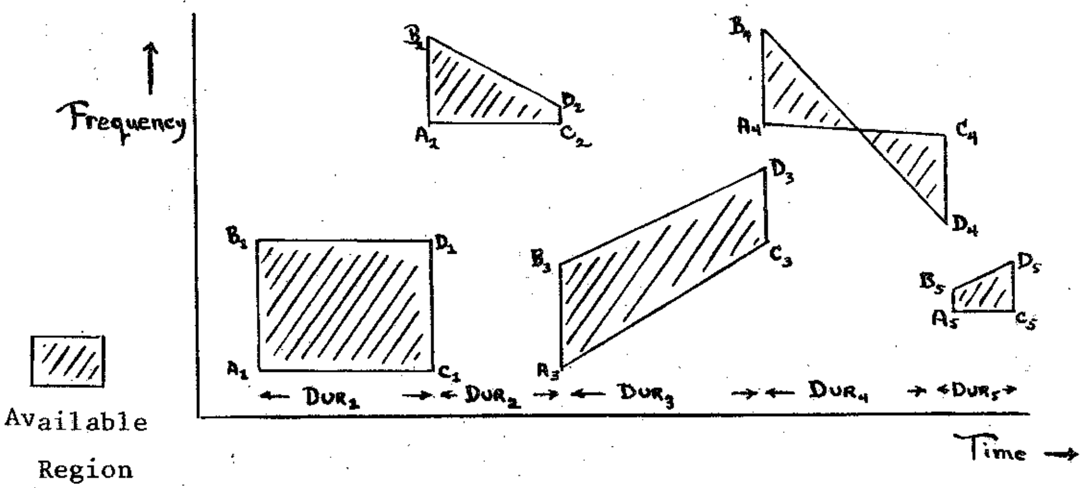


Fig. 1 Tendency mask for Poisson distribution

Each segment of the mask consists of two vertical lines, and two other straight lines connecting them, which may cross each other; thus, each segment of the mask is specified by five numbers. The integer data input expected by the program for this calculations is:

1) number of mask segments (maximum 10)
2) total number of events in the distribution
3) initial density in sounds per second
4) mask data: for each segment: A,B,C,D (in Hz.)
   DUR (in hundredths of sec.)

The calculation produces frequency-time co-ordinates for each point sound event. As well, a waveform number is chosen for each event (see Section III) and optionally a spatial position (see Section IV). These four values are coded into two data words and stored on dectape. The coding in the 18-bit binary configuration places the waveform number in the 5 high-order bits, and the frequency in the 13 low-order bits. A similar procedure is used to code the spatial position number and time coordinate respectively. With POD5 & 6, the waveform number is the object number.

	Waveform number	Frequency
	← 5 bits →	← 13 bits →
range:	(1 to 31)	(1 to 8191)

Figure 2: Data Coding

During the first calculation of the theoretical distribution, information regarding its sound densities and average time delays at the beginning and end of the entire mask is printed on the teletype. After the distribution is calculated, the user may at any time ask for an analysis of the structure in terms of sound densities and average time delays, taking into account <u>all</u> performance variables affecting the time field, as well as the clock setting being used. This information includes the density and average time delay as it occurs on the average over each mask segment, as well as representative samples of these variables taken from the first 10% and last 10% of the time duration of each mask segment. This information shows the user the effects of the frequency tendency mask, and, as well, all performance variables on the distribution and may allow a better choice of envelope data, performance speed or distribution characteristics to be made.

The user also has a choice of linear or logarithmic frequency scale for each individual mask segment; that is, giving each octave equal weight in the logarithmic case, or else making each integer frequency equally probable in the linear case, thereby resulting in a predominance of higher frequencies.

II. WAVEFORM INPUT AND STORAGE : (POD4 & 5)

The basic material from which the sound events are synthesized in real time by the computer are sets of data, either stored on dectape (as with POD4 & 5), or else calculated as a permanent array by the machine (POD6). Each array is a set of numbers describing the voltage variation required to produce one period of a periodic sound.

Because of the three different synthesis methods used, one by each of the three programs, the data format varies as follows:

POD4: 5 versions of 50 sample waveforms, each version of different, but, in the normal case, increasing amplitudes.
Total number of samples: 250 (stored on 1 block of dectape)
Amplitude Scaling: $2048 \pm (4000/2^n)$, n = 5,4,3,2,1

POD5: 1 waveform of 500 samples (stored on two blocks of dectape)
Amplitude scaling: 2048 ± 2000. An array of 60 amplitude factors is used to produce attack and decay sections of the envelope. These are of the form: $2^{((n-1)/10)}$, n=1,60

POD6: 1 sine waveform of 512 samples, calculated by the program.
Amplitude scaling as with POD5 .

The D-A converters accept numbers in the range 0 to 4095 to convert to voltages in the range 0 to +10 volts. The waveform data has a zero line of 2048, which is converted to +5 volts., and the amplitude variations on either side of the 2048 level have a maximum value of $\pm$ 2000 corresponding to a maximum voltage variation of 5 volts, peak to peak.

With POD6, the waveform data is computed by the program, but in the other versions, the data may be obtained in the following ways.

1) Waveform Library: A library of waveform formulas is available within the program, able to write data in the appropriate format on dectape. Ten common waveforms are included in this library, and may be used either as sound material or else as input for other waveform operations. For POD4, a permanent "waveform library" is available with the 10 basic waveforms and 8 or more derived ones for the convenience of users.

2) Hybrid Waveform Construction:

New waveforms may be constructed from those already on dectape. In the case of POD4, using 5 different sections, each section may be obtained from another existing waveform, or else part of it, allowing the user to change amplitude and timbral components within the sound material, in this way varying from the single basic waveform in increasing amplitude format. In POD5, where only a single waveform is used, it may be constructed from parts of other waveforms. The procedure is of an experimental nature, since it is difficult to organize harmonic content with any accuracy or predictability by this method. It has, however, served as an intermediary method until dynamic sound synthesis was achieved with POD6. To use the procedure, one is asked to specify waveform number, if it has sections (POD4) which one (1 to 5), the number of samples to be used and whether they are to be read in forwards or backwards (in the latter case one ends with the start sample number). After every group of samples is added, one may accept or reject it, and if one is working with sections, at the end of each one it may be accepted or rejected.

3) Waveforms obtained from other sources:

i) By using the first part of a Fourier analysis program developed at Utrecht by C.A.G.M.Tempelaars, one can read samples of live sound into the computer via the A-D converters. The samples, taken every 40 microseconds, are stored on disk and may then be retrieved by means of a sampling program that allows one to select part of the data to have transferred to paper tape. Then, using an option of POD4 or 5, these samples may be read in and converted to the format appropriate to the program. In POD4, each section is read in separately, or else derived from the same one. While complex sounds cannot be reproduced in this way, single waveforms of a periodic nature can be accurately transferred to the program. Any number of samples in any format of integers may serve as input to the program.

ii) Using another part of the Fourier analysis program at Utrecht, waveforms may be synthesized by specifying amplitude values and phase angles of the various partials. The output, after being heard, may then be placed on paper tape and transferred as already indicated. A somewhat simpler method of harmonic construction is included within the program itself and is described next.

4) Construction of a waveform with Harmonic content:

In many cases, the composer is not in possession of specific data for amplitudes and phase angles that is needed to construct properly a waveform by Fourier synthesis, or else he may wish to try various simple combinations first without going through the entire procedure of calculating Fourier values and having them transferred to this program. Therefore the possibility has been created to make a simple harmonic construction based on a waveform already existing on dectape.

In POD4, the 5 amplitude versions provide five equally spaced amplitude values which can be used as the amplitudes of the various partials, whereas with POD5, the user may specify an arbitrary integer scale, say 0 to 10 or 0 to 100, to indicate the relative strengths of the partials. A phase angle in degrees is translated by the program into a start sample number, considering 0 degrees to correspond with the first sample of the waveform. The values of the harmonics can be approximated by skipping a number of samples before taking the next one, the size of the skip being equal to the number of the partial. In POD4, with the 50 sample format, the fundamental and first four harmonics are added in this way, whereas in POD5, having a larger number of samples available, the user may specify the number of partials to be added, up to 50. In the case of 50 partials with POD5 or 5 with POD4, the number of samples per period drops to 10, so none are added above that point. For example, with 50 samples to describe a waveform, the values of the partials are found by adding together the samples denoted by the following groups of sample numbers taken vertically.

Sample Numbers of Values for 1 Period

Fund.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	......	47	48	49	50
1st.H.	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	......	43	45	47	49
2nd.H.	1	4	7	10	13	16	19	22	25	28	31	34	37	40	43	46	......	39	42	45	48
3rd.H.	1	5	9	13	17	21	25	29	33	37	41	45	49	3	7	11	......	35	39	43	47
4th.H.	1	6	11	16	21	26	31	36	41	46	1	6	11	16	21	26	......	31	36	41	46

An extension of this method is used with POD5. Since any waveform may be used as source material for this construction, more complex harmonic possibilities may also be realized with this simple method.

III. FREQUENCY MODULATION SYNTHESIS: (POD6)

The synthesis method incorporated in POD6 is a distinct break with that of the earlier programs in that, for the first time, a dynamic spectrum is achieved by means of a frequency modulation method developed digitally by John Chowning of Stanford University. The basic principle of synthesis involves a carrier sine wave, frequency c, modulated by another sine wave, frequency m, amplitude d. The modulation index I is defined as the ratio of d/m, and its value controls the strength of the pairs of upper and lower sidebands produced about the carrier frequency, each separated by a frequency distance equal to m. The basic spectrum generated is of the form:

$$/ c \pm n \cdot m / \qquad n = 1,2,3,\ldots \text{ for nth order sidebands}$$

It should be noted that both harmonic and inharmonic spectra are produced in this way, depending on whether the c/m ratio is in a simple integer proportion (e.g. 1:1, 2:1, etc.) or else involves rational values (e.g. 5:7, 4:13, etc.). The absolute value signs signify that if the lower sidebands are negative, they are reflected back into the positive domain with a phase inversion. The amplitude of the nth order sideband is given by the nth order Bessel function $J_n(I)$, where I is the modulation index.

If I is made to vary during the course of the sound, then the amplitudes of the sidebands will also change and the resultant spectrum will be time-dependent. Large values of I produce strong high-frequency components, whereas low values result in small deviations from the basic sine wave. The correlation of modulation index with envelope is crucial, and in the present program it is hoped to implement various possible configurations; the principal one thus far involves a growth of the modulation index from 0 to the maximum value specified by the user during the attack, then a decay of I during the steady amplitude state, and finally I remaining constant during the amplitude decay, acting as a kind of reverberation. This type of spectral change is characteristic of many familiar instrumental and percussive sounds and the present method (discussed in more detail in Appendix B, Section III) allows a simple and yet comprehensive control of timbre in a way not possible in the previous programs.

IV. OBJECT SPECIFICATION AND SELECTION:

The description of the material comprising a single sound event as calculated in a Poisson distribution is called an object. With POD4, objects as such are not actually referred to, but rather, waveforms are chosen (up to 10 in number), and for each an envelope is assigned. For the other programs, the object definition is as follows:

POD5: i) waveform
ii) envelope values (3 numbers, describing the attack, steady state and decay times in hundredths of seconds)
iii) amplitude modulation frequency

POD6: i) envelope data
ii) c-m ratios (integers up to 500)
iii) maximum modulation index

Maximum number of objects: 30.
Restrictions: Only 5 different waveforms may be used with POD5

For each compositional section, the waveforms or objects to be used are chosen by specifying: 1) the number of objects or waveforms, and 2) their identification numbers. Ten waveforms with POD4, and 30 objects with POD5 & 6 are possible over the course of the entire composition.

The objects or waveforms are assigned to each event calculated in the Poisson distribution by various statistical means. These are:

1) ALEA: an aleatoric selection is made from those stated as available, i.e. equal probability
2) RATIO: a weighted aleatoric selection is made, based on whole number ratios specified by the user for each object
3) SEQUENCE: a simple fixed sequence is made, based on the order given by the user with no statistical fluctuations.
4) TENDENCY: a tendency mask, similar in form to that for the Poisson distribution is specified as in Fig. 3, describing not only which waveforms or objects are available at any given time point, but also the weighted ratios by which an aleatoric selection from them will be made. This selection then can be regarded as a set of selection ratios shifting in time. Note that the co-ordinates A,B,C,D are expressed in integer percentages. Although the distinction has not been made entirely explicit in this text, the use of a tendency mask for selection should not be confused with its use in the Poisson calculation where it merely delimits an available area. Other than that it

plays no part in the actual selection process which is entirely determined by the Poisson method (see Appendix A). Whereas it serves a useful purpose for object selection in that we can say at such and such a time so many objects are available with certain ratios weighting their occurrence probability, the tendency mask as a selection principle in itself for the time and frequency structure is foreign to the basic philosophy of the Poisson approach used here, where the density at any given time is affected by the characteristics of the entire distribution. It presupposes that the composer is concerned with such overall aspects as density changes from which the small scale time structure is determined, instead of only making short-range selection at any given time moment unrelated to all others as implied by the use of tendency masks as a selection principle.

If the sum of the segment durations for this mask equals the total duration of the Poisson mask, then there will be a one-to-one correspondence between their respective time values. However, if they are not equal, the object or waveform tendency mask durations will be scaled so that they fit the Poisson duration. Therefore, it is possible simply to give the object or waveform mask durations as proportional integers (e.g.3:2:1:5) and let the program scale them relative to the Poisson mask length.

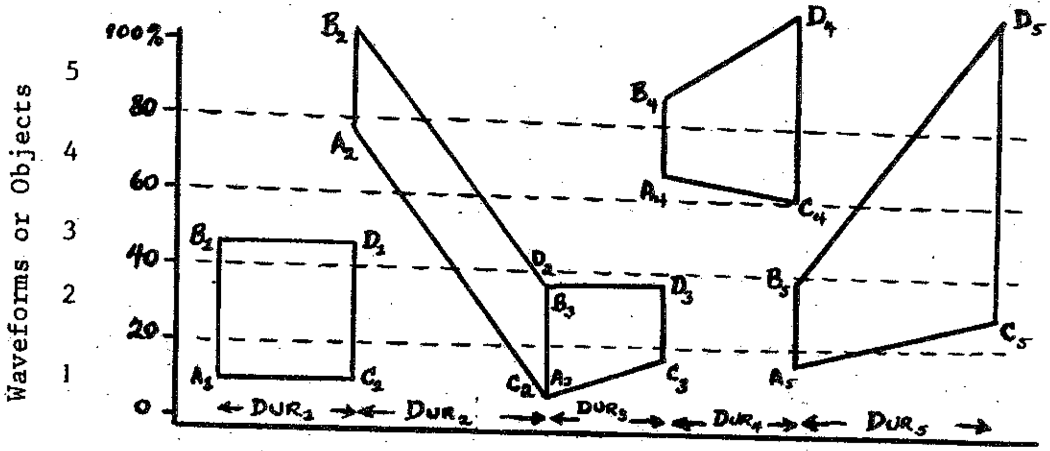


Figure 3. Waveform or Object Tendency Mask

5) ALEA with Synthesis Check (POD6 only): Because of the upper limit on the possible modulation frequency in POD6, it is possible that a carrier frequency will be calculated that cannot be synthesized correctly for the given c-m ratios given by the user. Therefore, a fifth selection mode has been included in which, for every calculated frequency point, the program checks to see how many of the available objects can be properly synthesized at that frequency, and then makes an aleatoric selection from only them, or else, if none are available, it chooses the nearest one.

V. SPATIAL POSITION: (POD4 & 5)

The sound events may appear to be located at any of seven positions between the left and right speakers, based on the illusion created by the presence of unequal amplitudes of the same sound in each speaker. At present no phase relationships are created as well, although this would be a possible extension. If no stereo option is specified, the synthesis occurs monophonically. With POD4, the upper two-channel limit is 2400 Hz., compared with the monophonic limit of 6400 Hz. but with POD5, a smaller number of samples per period are used in the upper region, namely 5 per period for single channel output at 8K and 4 per period at 6K for two-channel sound. Therefore, is it possible to use either mode for frequencies up to 6000 Hz. with POD5.

Similar to the procedure for object selection, various statistical methods may be used for the selection of spatial position for each object, as follows:

1) ALEA: an aleatoric selection for each object is made from the 7 available positions.
2) FREQUENCY CORRELATION: the frequency range is divided into 7 octaves as follows:

Octave 1:	50 -	100 Hz.
2:	100 -	200 Hz.
3:	200 -	400 Hz.
4:	400 -	800 Hz.
5:	800 -	1600 Hz.
6:	1600 -	3200 Hz.
7:	3200 -	6400 Hz.

Two correlation methods are possible, each assigning these octaves to specific spatial positions:

1) Linear: position 1 is assigned automatically to octave 1, position 2 to octave 2, and so on for all seven.
2) Ordered: the program asks the user for an ordered assignment of positions. Note that this allows one to assign the same position for more than one octave.

Example: 1 7 1 7 3 5 4

assigns: position 1 to octaves 1 and 3
position 7 to octaves 2 and 4
position 3 to octave 5
position 5 to octave 6
position 4 to octave 7

3) TIME CORRELATION: by means of a tendency mask, similar to that for object or waveform selection, the available spatial positions are mapped out and an aleatoric selection is made from those available at each time point. The format of data input for this mask is identical for that of the waveform or object selection tendency mask (See Fig. 3), except that the vertical scale will always be divided into seven parts corresponding to the seven spatial positions.

The spatial effect is achieved by having the amplitude of the event reach different maximum values in the two channels, choosing from the 5 in POD4 and the 60 in POD5. For each position, one of the channels at least will reach its maximum amplitude, with the other less than or equal to it.

Spatial Position		Maximum Amplitude Section Reached	
	Channel	POD4	POD5
LEFT			
1	L	5	60
	R	2	30
2	L	5	60
	R	3	40
3	L	5	60
	R	4	50
4	L	5	60
	R	5	60
5	L	4	50
	R	5	60
6	L	3	40
	R	5	60
7	L	2	30
	R	5	60

Figure 4. Maximum Amplitude Section as a Function of Position

The procedure in POD5 is much more successful of course, since the envelopes are smooth and the distinction between positions more obvious. In both cases, however, a spatial movement is involved, since the sound begins in the centre and moves to one of the seven positions during the attack and returns during the decay. The apparent speed then also varies for each position. With hybrid waveforms in POD4, the various sections will appear at different times in each channel, so if a series of different timbres or amplitudes has been constructed, they will be presented in a different order for each position.

In general, the amplitude control with POD5 could be developed for other purposes, such as a selection of the maximum amplitude of an event, or a more complex envelope control, as well as amplitude modulation.

VI. PERFORMANCE VARIABLES: AMPLITUDE-TIME CHARACTERISTICS:

The interpretation of certain time and amplitude characteristics of the performance of the distribution of sound events is specific only to the operation of the synthesis program; therefore, it is independent of the calculations made for the Poisson distribution, and can be regarded as its realization. The variables that affect these characteristics can be changed by specifying their new value, without any recalculation of the distribution. By implication, then, these variables are not statistically, but rather perceptually determined by the user. In many ways, the variety of possible combinations that may be tried will have perhaps more effect on the psychoacoustic nature of the sound structure, at least in its overall character, than will certain variations of the Poisson calculation. Because these aspects can be changed quickly, the composer can spend some time adjusting their values to suit his purpose, as opposed to trusting some pre-arranged numbers that he hopes will produce the desired effect. The variables that control these characteristics are as follows:

1) Speed of the Time Structure:

The real-time clock used to interpret the time coordinate is controlled by an external generator whose frequency setting is manually adjusted. Normally, however, it is assumed to run at 30 KHz., with one cycle being 1/30,000 sec. Since the speed of the time structure has been specified and calculated in terms of hundredths of seconds, the time coordinates are normally multiplied by 300 to convert their values to the clock's speed. That is, when the clock counts 30,000 per second, the value of a second being used (i.e. 100) is related to it by a factor of 300.

Unless changed by the user, the value of 300 will be assumed. However, by using the VI option (see Section VII for a discussion of the control options), the speed can be reset to any other factor, as long as the number of clock units asked for does not exceed the computer's negative integer capacity. For longer time delays than two seconds, say, it is necessary to set the clock to a lower frequency and adjust the speed factor accordingly. The normal factor = clock frequency/100. In most cases, it seems better to adjust performance densities by means of the speed factor, instead of changing the theoretical densities of the distribution. Since both frequency and envelope values depend on machine cycles for timing they are not affected by a change of clock speed, the effect being similar to a tempophone used for constant frequency and variable time.

2) Object Data:

Object data may also be thought of as performance variables, since this information is not explicit in the calculation of the Poisson distribution, and, as a result of that distinction, may be changed without the necessity of recalculation, allowing the results to be heard immediately. This data includes:

Program	Data Type	Changed By
POD4	1) envelope data	V2 option
POD5	1) waveform	V2 option
	2) envelope data	V2 option
	3) amplitude modulation frequency	V5 option
POD6	1) envelope data	V2 option
	2) c-m ratios	V5 option
	3) maximum modulation index	V5 option

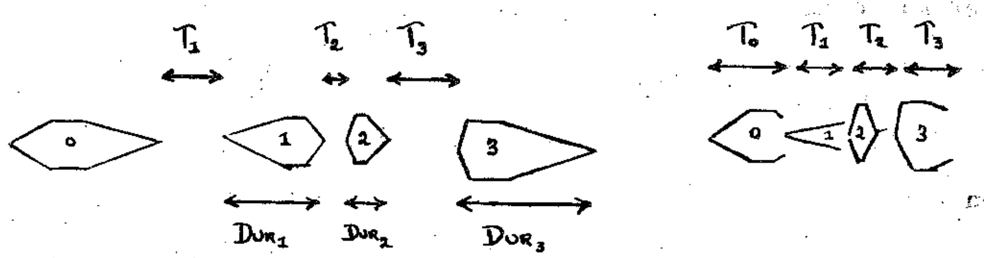
In POD5, the waveform may also be regarded as a performance variable since during the calculation only the objects are selected by number, without regard to what their definition may be, unlike POD4, where waveforms themselves are chosen by number. Therefore, with POD5 & 6, all object data may be changed without a recalcultion. As well, in POD5, if one is using the maximum number of waveforms, and then would like to change one of them, the program offers the possibility of replacing all occurrences of one waveform by a new one, still keeping the total number within the limit.

In all of the programs, the envelope data comprises 3 numbers in hundredths of seconds describing the attack, steady state and decay times of the amplitude change. In POD6, the c-m ratios may be any pair of numbers from 1 to 500 (and also 0:1). Thus, one may use simple ratios, such as 1:1, or else small deviations from them, such as 100:101 or 499:500, causing the lower and upper sidebands to be slightly different, resulting in beat frequencies.

3) Relationship between Entry Delay and Duration:

This variable, more than any other, is key to the interpretation of the time structure calculated by the Poisson distribution. Essentially it determines whether the duration of the event, as specified by envelope data, has priority over, or is dependent on the entry delay between events. These two situations are illustrated in Figure 5. In the so-called 'dependent' case, the next entry only occurs after the entire duration (Env_i), as well as the time delay derived from the Poisson time structure (T_i), have elapsed, as on the left. In the 'independent' case, the clock time, representing the duration between successive time co-ordinates calculated by the Poisson procedure, has precedence, and the duration of the envelope is allowed to continue only as long as the clock is counting. In the case where a very dense Poisson distribution has been calculated, the independent relationship will reflect this density, and if the resulting rapidity of sound events all running into each other forming a kind of cluster effect is not desirable, it will be necessary to slow the time structure by means of the performance speed factor, or by resetting the clock.

The choice between these two modes theoretically represents the user's choice of the hierarchy between entry delay and duration of envelopes as to which takes precedence. On a practical level, the slower, dependent case is useful for checking the realization of the distribution in a situation where the strict interpretation of the time structure, represented by the independent case, creates too dense a performance for the user to understand the exact compositional nature of the result. Note, too, that the effects of which time mode is being used will be taken into account when an analysis of densities in the distribution is made with the D2 option.



(Dependent Case) (Independent Case)

Figure 5. Relations between Duration and Entry Delay.

3) Amplitude Modulation:

With POD5, the additional performance variable has been added of amplitude modulation, where the modulating frequency may either simply create more complex amplitude variation superimposed on the envelope (with frequencies from 8 cycles per sec. to those less than 1 per sec.), or else affect timbre by creating frequency side-bands (with modulating frequencies from about 8 or 10 Hz. up to a maximum of 1/8th the carrier frequency).

VII. PERFORMANCE AND SOUND SYNTHESIS:

The technical details of the strategies involved in the synthesis of fixed waveforms at various frequencies, as well as the frequency modulation method, are discussed in the various sections of Appendix B. The following is a list of performance options currently available.

P1) A monophonic 'test' performance of the distribution with a simple triangular waveform and short percussion-like envelope. Although the speed of performance is variable, only the 'dependent time mode' is possible, as shown in Fig. 5. The program performing this synthesis does not use the strategy described in Appendix B, but rather a simpler use is made of basic machine operations (adding, subtracting, comparison, etc.) to produce these single waveform sounds, limited in range from 60 to 1500 Hz.

P2) Normal performance of the calculated distribution with waveform or object selection, and optional two-channel output. The full range of performance variables described in Section VI apply to this synthesis, whose technical description appears in Appendix B.

P3) Waveform or single sound test. A single sound event in the synthesis mode of the program may be heard by specifying its parameters. For POD4 & 5, this means the waveform number, envelope, frequency and for POD5, amplitude modulation frequency. For POD6 the variables are envelope, c-m ratio and maximum modulation index.

P4) Performance of all compositional sections including the current one.

P5) POD4 & 5: Two-channel waveform test. Any waveform stored on dectape may be heard in a spatial position between the two speakers by specifying its number, frequency, envelope and spatial position.

POD6: Repeated sound test. This object test is similar to P3 except that the sound repeats until terminated by setting an accumulator switch.

These five options correspond to the 'P options' listed in Appendix C. As well, the 'R options', designated R1 and R2 imply not only calculation of the distribution but also, following that, the performance specified by P1 and P2 respectively. With R3 only calculation is done.

VIII. THE CONTROL PROGRAM 'SAM':

This program acts as a combined switchboard and monitor for all of the operations of the program, allowing the user to make the choice of the area where he would like to work. As well, it collects any additional information needed by the program to carry out specific operations. The user directs the flow of control of the program by means of a code consisting of an alphabetic character and an integer from 1 to 5. Groups of related operations are given the same letter, and in some cases, there is a correspondence between the digit meaning with one letter and that of another.

The operations already discussed is Sections I to VII are referred to by the alphabetic codes;

Section I: Poisson Distribution: R & D options
Section II: Waveform Input: F Options (POD4 & 5 only)
Section III: Frequency Modulation information: O options (POD6)
Section IV: Object and waveform Selection: W options
Section V: Spatial Distribution: S options (POD4 & 5)
Section VI: Performance Variables: V options
Section VII: Performance and Synthesis: P options

A list of the specific options appears in Appendix C. Other options involving various representations of the data involved on teletype, paper tape, or graphic display have also been included, as well as one to change specific values of the Poisson data. These are:

1) Change data for Poisson distribution: C options
2) List data for Poisson distribution or objects: L options
3) Graphic display of distribution, masks, waveform data: G options
4) New data tapes, waveforms data, co-ordinates: T options

Many provisions have been made to avoid user mistakes becoming unrecoverable in the program. Often, once a mistake has been made, another option may be used to correct it. The program indicates its readiness to accept a command by issuing the symbol ! and if an unrecognizable command is made after that, it is ignored and the symbol is re-issued.

IX. SECTION STORAGE:

A composition in the sense of these programs may be built up of 10 sections. Once one is done, with all changes made and the data stored, the user can move on to the next section and repeat the process of selection and calculation. At any time the accumulated data may be played back for inspection up to and including the current section, by requesting the P4 option. At the end of each section a detailed list of all the data used in that section is printed on teletype for future reference or reconstruction. The protocols generated by the program form a unique record of compositional working such as is seldom found in areas of creative thought, and their study may produce theoretical insights into the compositional process that has occurred.

Since the performance variables apply only to a specific section, they are stored in order to be put into effect when the section is replayed. The 10 waveforms in POD4, or the 30 objects of POD5 & 6, may be used in more than one section, but changing any of their parameters (e.g. envelopes) after they have been used previously will change the result of the earlier section when reheard. Since the spatial position is coded with the time coordinate, it is possible for some sections to occur monophonically and others with two-channel output. In deciding matters of time structure it should be remembered that the speed factor (specified by V1) may be different from one section to another, but that the clock setting is manual and should be left at a decided value for the entire composition.

Two additional programs have been written to allow the user to store the data being used on dectape as permanent files on disk, then transfer them to a private dectape, and retrieve them at a later time for further work. Only completed sections may be stored in this way. All data in core memory at run time that is essential to continue work is (optionally) put on paper tape at the end of each section. With its use, it is possible to restart the program and begin work on the next section after those already completed. Because of the recording structure of the dectape system, it is not as yet possible to rewrite any block of data except the last without erasing other blocks, but it may be possible later to incorporate an editing facility to allow parts of completed compositions to be reworked. Also, without mass data storage by digital tape, it is not yet possible to mix sections, but this is being done at present with two and four channel audio tape recorders.

X. CONCLUSION: "An Almost Final Word":

The experience gained in the last year with these programs from their inception, development and use leads me to believe that there should be no final word about them, but that rather the programs must continue to grow as our insight into the complexities of programmed musical intelligence develops, an insight to which it is hoped their use will continue to contribute.

The emphasis must be on flexibility and learning. Such programs need to be as accessible as possible to the user, helping him to realize his ideas easily and at the same time deepen his insight into the process in which he is involved. The importance then of running these programs on the smaller machines that are now increasingly available, and allowing ample time for working interactively with them, is obvious. At the level of understanding the structure of sound material, as well as working with the larger organizational aspects of sequences of these sounds, a unique kind of learning situation is possible with this type of program. However, the programs too must always be open to change and improvement or else the insight gained from their use is partly wasted. The author has been fortunate in this respect in being able to benefit from the valuable suggestions of the various users of the programs this past year at the Institute of Sonology, most notably Dr. Otto E. Laske, Mr. Don Cardoza, and Mr. David McGuire, to whom thanks are due.

It should be emphasized that, although a heavy reliance has been made in these programs on statistical procedures, other compositional ideas can and should be implemented in this way. When used in programmed form, these ideas are subject to the kind of close scrutiny under which they can best develop. The statistical method, however, has proved its worth by allowing a greater emphasis to be placed on abstract planning at a higher, syntactical level, offering a basic facility by which the smaller details could be taken care of accordingly. As well, it has allowed an emphasis on characteristics such as available areas, densities and sound objects which seem at least a step above the pre-occupation with sound parameters that threatens to subvert the recognition of the unity of perceptual experience. Much work still needs to be carried out in this direction, and it is hoped that the present efforts may stimulate such development.

Barry Truax,
Institute of Sonology,
Utrecht, Netherlands.
May 28, 1973.

APPENDIX A: CALCULATION OF THE POISSON DISTRIBUTION:

Poisson distribution is the name given to the random distribution of events which occur as single units, as opposed to a continuously distributed variable. Two conditions are assumed to hold:

1) The occurrence of any one event is independent from that of any other, and no particular event is *a priori* more likely than any other. In other words, time and space are homogeneous. (The musical implications of the use of statistical distributions is discussed in the Preface).

2) The density of events is not to be too large. This condition represents the area where the Poisson distribution shades over into the Gaussian or normal distribution of a continuous variable, around a density of 10. In terms of sound events, this limit seems no more restrictive than that of the ear's limit to distinguish separate sound events.

I. THEORY:

The basic Poisson theorem gives the probability of finding k events in an interval or area where the density of events is known to be λ, that is, the average number per unit area.

$$\lambda = \frac{T}{N}$$

where: T is the total number of events
N is the number of units or area over which they occur.

The probability p_k of finding k events, given a density λ is:

$$p_k = e^{-\lambda} \cdot \frac{(\lambda)^k}{k!} \qquad \ldots . (i)$$

note: $k! = 1.2.3. \ldots .(k-1).k$

Once the density λ is known, this theorem can be used to predict the probability of k events happening in that area.

Secondly, in order to determine where in the given space the events will occur, we need a second conclusion from the Poisson theorem. If the space under consideration is divided into small enough intervals that the probability of more than one event occurring in the same space is negligible, then we may ask what is the probability of finding (L-1) of these units with no event, and unit L with one event.

From equation (i), the probability of $k = 0$ in L intervals of unit length t, that is, in length Lt is:

$$p_0 = e^{-L\lambda t}$$

The probability of k greater than 1 has been said to be negligible; that is:

$$p_k = 0 \quad \text{for } k = 2,3,4, \ldots.$$

But the sum of all p_k must be unity:

$$\sum_k p_k = 1 = p_0 + p_1 + p_2 + \ldots$$

Therefore: $$p_1 = 1 - p_0 = 1 - e^{-L\lambda t}$$

Solving for L: $$L = \frac{-\ln(1-p_1)}{\lambda t} \qquad \ldots \text{(ii)} \qquad /\text{note: } \ln \equiv \log_e$$

One can interpret this equation in terms of a situation where one is 'travelling' over a certain interval in which the density of events is known. The distance one travels before 'encountering' an event is a variable, but according to the above equation, it is less probable for L to become large; that is, it is certain that one can travel zero distance without meeting an event, but it becomes less and less probable that one can travel a larger and larger distance without encountering one - in fact, the probability diminishes exponentially. In this analogy it does not matter if the 'distance' referred to is time or space. Some representative values of the solutions of equations (i) and (ii) are given in Tables 1 and 2. Note that for 100 units, the distance where the probability is 50% that one encounters an event is not after 50 units, but after 69. In other words, it is 50% probable that the event will not occur in the first 69% of the average distance.

II, APPLICATION TO A FREQUENCY-TIME DISTRIBUTION OF SOUND EVENTS:

In order to calculate a distribution of sound events that more or less conforms to Poisson's theorem, it is necessary to know only the total number of events desired (approximately) and the area over which they are to be distributed. The present calculation, however, includes two additional options thought to be of use to the composer.

1) Variable Density: The average density of the theoretical distribution may increase or decrease linearly within the entire distribution; this requires specifying an initial density. If d_n is the density in unit area n, and u is the fixed amount by which it increases or decreases from one unit area to the next, we may write the following set of equations:

$$
\begin{aligned}
d_0 &= d_0 \\
d_1 &= d_0 + u \\
d_2 &= d_1 + u = d_0 + 2u \\
d_3 &= d_2 + u = d_0 + 3u \\
&\vdots \\
d_n &= d_{n-1} + u = d_0 + (n-1).u
\end{aligned}
$$

Summing each side: $d_n = n.d_0 + u.(0 + 1 + 2 + \ldots + (n-1))$

Since d_n is the density per unit area, it is equivalent to the average number of events in that area. Therefore, $T = \sum_n d_n$, where T is the total number of events.

Therefore: $$T = n.d_0 + u.\frac{n.(n-1)}{2}$$

Solving for u: $$u = \frac{2.(T-n.d_0)}{n.(n-1)} \qquad \ldots \text{(iii)}$$

Therefore, by specifying n, the number of units over which the density changes, and d_0, the initial density, and T, the total number of events, the unit density increment may be calculated.

Note: if $T = n.d_0$ then $u = 0$ (uniform density)

if $T > n.d_0$ then u is positive (increasing density)

if $T < n.d_0$ then u is negative (decreasing density)

2) Tendency Mask. Secondly, a tendency mask (see Fig. 1) outlines what frequency areas are available at a given time, allowing the user to restrict the areas in which the distribution occurs. The requirement of homogeneity stated earlier is subject, then, not only to

restrictions imposed by the ear, but also to those made by the composer.

For purpose of calculation, the tendency mask, with its various segments, is translated into a series of rectangular areas each having the same area as the corresponding one of the mask, and the same duration as well. The area AR of a mask segment described by the co-ordinates A,B,C,D,DUR is:

$$AR = \tfrac{1}{2}.DUR.(B+D-A-C) \quad \text{for } D \geqslant C$$

$$\text{or:} \quad AR = \tfrac{1}{2}.DUR.\frac{(B-A)^2 + (C-D)^2}{(B-A) + (C-D)} \quad \text{for } D < C$$

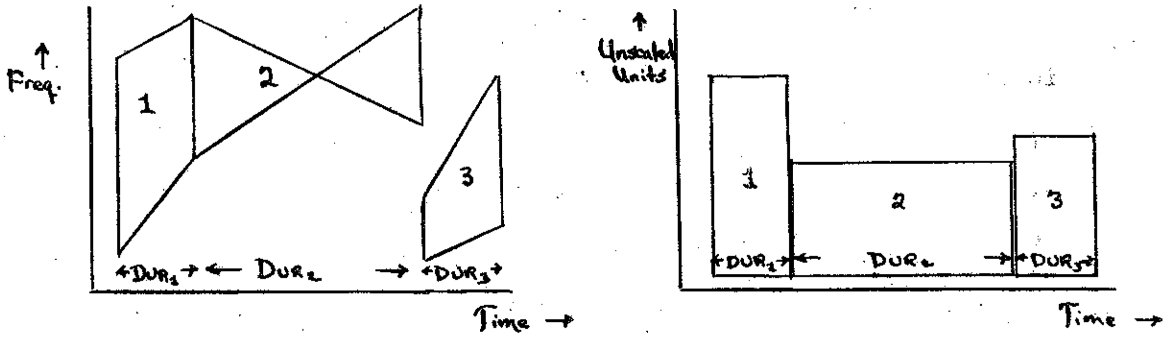


Fig. 6. Re-arrangement of Poisson Mask for Calculation.

The situation that results when the areas of the various segments are not the same area is that the apparent density will not be clearly increasing or decreasing according to the same pattern established by the specification of total number of events and initial density. However, it has been decided not to make any compensation for this alteration since it may be that the composer wishes to construct a tendency mask whose changing areas impose a counter-trend on the basic linearly changing density already established.

III. CALCULATION:

Operating from Fig. 6, each segment of the mask is divided into 10 sections of equal area, as in Fig. 7. If the durations of each segment of the mask were equal, then each of these smaller segments would represent the same length of time, but since this is rarely the case, the sound density is not regarded as the density in a unit, but rather as an absolute number of sounds per second, and the program translates this quantity into the density per small mask segment.

From the equation: $d_n = d_0 + (n-1).u$ for $n = 1,2,3,\ldots,10$

the density d_n is obtained for each subdivision of the mask, given the initial density d_0 at the beginning of the mask segment. This density in sounds per second, is again translated into the density in the small sub-segment of the mask, and equation (i) used to calculate, first for $k = 0$, then $k = 1$, etc. the probability p_k. Because p_k is unequal for different k, the random generator IALEA cannot be used directly, but rather, given a random number IR, the program compares it to the 'accumulated' probability P_k where:

$$P_k = p_0 + p_1 + p_2 + \ldots + p_k$$

until k is found such that: $P_{k-1} \leqslant IR < P_k$

This value of k is then taken as a random number of events for the section, given the weighted probability of the Poisson distribution. The same method is used for the selection principle Ratio when needed to change the equal distribution of a random generator into a weighted distribution by ratios.

To distribute these events, the areas of Fig. 7 are subdivided into very small units, as in Fig. 8, 1/100 sec. horizontally and 1 Hz. vertically in the linear case; in the logarithmic case, all frequency co-ordinates of the mask are put on a logarithmic scale, base 2 on octaves starting at 50 Hz. The density is determined by the values of the k events and the area AR/10, and the value of L is found from equation (ii) for the case of (l-1) units with no event and the Lth unit with one event. The units available according to the tendency mask are counted, bottom to top, left to right, at 1/100 sec. intervals, and the appropriate frequency-time co-ordinate noted.

Once these k points are so placed, then if the masked area is not yet filled, the program goes back to equation (i) and selects another value of k events, thus allowing statistical variations in density. By subdividing each mask segment into 10 sections, the density slowly changes across the segment, without violent jumps at the boundaries, except those caused by changes in apparent density because of differences in area of adjoining mask segments.

For the purposes of calculation it does not matter what in fact the event will be, since it is merely regarded as a point event. The theory then stops at a elementary level and sonological decisions must then take over to give the events musical function.

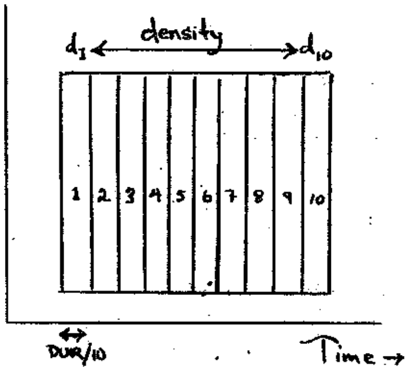


Figure 7. Division of Mask Segment

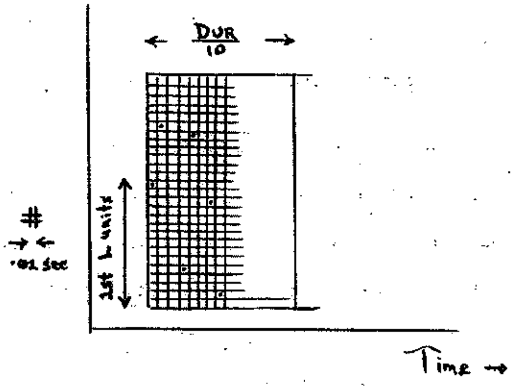


Figure 8. Small Unit Subdivision

TABLE I: Representative Values for the Solution of the Poisson Equations:

Equation (i): $$p_k = e^{-\lambda} \cdot \frac{(\lambda)^k}{k!}$$

density	percentage probability p_k for various k								
d_n	k = 0	1	2	3	4	5	6	7	8
0.5	60.65	30.33	7.58	1.26	0.16	0.02	0.00	0.00	0.00
1.0	36.79	36.79	18.39	6.13	1.53	0.31	0.05	0.01	0.00
1.5	22.31	33.47	25.10	12.55	4.71	1.41	0.35	0.08	0.01
2.0	13.53	27.07	27.07	18.04	9.02	3.61	1.20	0.34	0.09
2.5	8.21	20.52	25.65	21.38	13.36	6.68	2.78	0.99	0.31
3.0	4.98	14.94	22.40	22.40	16.80	10.08	5.04	2.16	0.81
3.5	3.02	10.57	18.50	21.58	18.88	13.22	7.71	3.85	1.69
4.0	1.83	7.33	14.65	19.54	19.54	15.63	10.42	5.95	2.98
4.5	1.11	5.00	11.25	16.87	18.98	17.09	12.81	8.24	4.63
5.0	0.67	3.37	8.42	14.04	17.55	17.55	14.62	10.44	6.53
5.5	0.41	2.25	6.18	11.33	15.58	17.14	15.71	12.34	8.49
6.0	0.25	1.49	4.46	8.92	13.39	16.06	16.06	13.77	10.33
6.5	0.15	0.98	3.18	6.88	11.18	14.54	15.75	14.62	11.88
7.0	0.09	0.64	2.23	5.21	9.12	12.77	14.90	14.90	13.04
7.5	0.06	0.41	1.56	3.89	7.29	10.94	13.67	14.65	13.73
8.0	0.03	0.27	1.07	2.86	5.73	9.16	12.21	13.96	13.96
8.5	0.02	0.17	0.74	2.08	4.43	7.52	10.66	12.94	13.75
9.0	0.01	0.11	0.50	1.50	3.37	6.07	9.11	11.71	13.18
9.5	0.01	0.07	0.34	1.07	2.54	4.83	7.64	10.37	12.32
10.0	0.00	0.05	0.23	0.76	1.89	3.78	6.31	9.01	11.26

TABLE II: Representative Values for the Solution of the Poisson Equations:

Equation (ii): $L = -\ln(1-p_1)/(\lambda t)$

For a given density d_n per 100 units, the number of units L with probability p_1

density	p_1 = 10%	20%	30%	40%	50%	60%	70%	80%	90%
1.0	230.26	160.94	120.40	91.63	69.31	51.08	35.67	22.31	10.54
2.0	115.13	80.47	60.20	45.81	34.66	25.54	17.83	11.16	5.27
3.0	76.75	53.65	40.13	30.54	23.10	17.03	11.89	7.44	3.51
4.0	57.56	40.24	30.10	22.91	17.33	12.77	8.92	5.58	2.63
5.0	46.05	32.19	24.08	18.33	13.86	10.22	7.13	4.46	2.11
6.0	38.38	26.82	20.07	15.27	11.55	8.51	5.94	3.72	1.76
7.0	32.89	22.99	17.20	13.09	9.90	7.30	5.10	3.19	1.51
8.0	28.78	20.12	15.05	11.45	8.66	6.39	4.46	2.79	1.32
9.0	25.58	17.88	13.38	10.18	7.70	5.68	3.96	2.48	1.17
10.0	23.03	16.09	12.04	9.16	6.93	5.11	3.57	2.23	1.05

APPENDIX B: SOUND SYNTHESIS METHODS:

SECTION I: FIXED WAVEFORM SYNTHESIS (POD4):

The basic concerns for sound synthesis in real time of periodic sounds with the computer are:

1) The time taken to output one sample through the D-A converter and obtain the next value, preparing the way for its output. The inverse of this time is the maximum sampling rate, whose value affects the sound quality.

2) In order to produce various frequencies, one must either vary the time between samples, or else the number of samples per period, that is, one may use either a fixed or variable sampling rate. In general, variable sampling is used by these programs, but the number of samples per period is made to vary as well as the time between samples.

3) The gradation of discrete frequency levels produced should be as fine as possible to minimize the error between the frequency requested and that produced.

4) If the sound is to be periodic with a variable amplitude, the program must allow for the data produced during one period to be repeated, with, at certain times, a change of amplitude as well. However, the time lag between periods must not become a significant delay with respect to the period of the sound, or else distortion will result.

If N samples comprise a single period of a sound, and if the delay between samples is dt, then the overall period of the sound, T, related inversely to its frequency, F, may be expressed by the equation:

$$N \,.\, dt = T = 1/F \qquad \ldots(1)$$

$$\text{or:} \quad F = 1/(n.dt) \qquad \ldots(2)$$

From equation (2), it can be seen that either a large N or large dt will severely limit the maximum frequency possible. Also, if F is to be variable, then either N or dt or both must be varied. The strategy adopted in the synthesis method of POD4 is to make dt the major variable, with three values of N taken in the different frequency regions.

The basic machine cycle of the PDP-15 computer is 0.8 microseconds (0.8×10^{-6}), and so it provides the smallest unit of time variation with which we can work. Also, it should be noted that the real-time clock of the computer could be used for this purpose, but since it normally runs in the range of 30 to 50 KHz., its basic cycle is 20 to 33 microseconds, and therefore is not as suited for the purpose of optimizing the output frequency. Instead, it was decided to use the real-time clock for the timing between sound events, and rely on the basic programming speed to control the micro-structure of the synthesis result. The implication of using the fixed time rate of the machine cycle to control frequency and also envelopes, which derive their timing by counting periods, and the variable time of the real-time clock to control the larger time structure of the sound events, is to allow the time structure to be varied without affecting the frequency or amplitude values, similar to a tempophone being used to speed up or slow down sounds on tape without affecting their pitch.

The fastest possible loop satisfying requirement (1) was found to be 9.0 microseconds. If requirement (2) is added, that is, allowing the time between samples to be a variable, the fastest program loop was found to be 11.4 microseconds, with the variation being in units of 1.6 microseconds. Since the basic machine cycle is 0.8 microseconds, two such loops could be made, running 1 cycle apart in speed. This may be expressed by the equation:

$$dt = 11.4 + 1.6\,T + 0.8\,n \qquad \ldots(3) \text{ microseconds}$$

where T is the timing variable ($T = 0,1,2,3,\ldots$)
and n is a secondary timing variable representing the time difference between two loops differing by only one cycle ($n = 0,1$)

Combining equations (1) and (3) gives:

$$11.4 + 1.6\,T + 0.8\,n = \frac{10^6}{N.F} \qquad \ldots.\ (4) \text{ microseconds}$$

From (4), with $T = n = 0$,

$F_{max} = 1755$ Hz. for $N = 50$ samples
$F_{max} = 3510$ Hz. for $N = 25$ samples
$F_{max} = 5150$ Hz. for $N = 17$ samples

Since the maximum frequency using 50 samples was only 1755 Hz., it was decided to use two other values of N (i.e. 25 and 17) to extend the upper range, producing maximum frequencies as indicated. In addition, three faster loops, again differing by a cycle, but allowing no variation in time within the loop were added to produce three very high frequencies; the equivalent expression to (4) then would be:

$$9.0 + 0.8\ n = \frac{10^6}{N.F} \qquad \ldots(5)$$

where n = 0,1,2 (note: n = 3 is equivalent to equation (4) with T = n = 0)

For N = 17 samples: F = 6540 Hz. for n = 0
F = 6000 Hz. for n = 1
F = 5550 Hz. for n = 2

As mentioned earlier, the discrete frequency values produced by the fastest timed loop with N = 50, 25 and 17 samples tend to be somewhat far apart. Therefore, not all of them are used in the case of N = 50, but rather, above an arbitrary level (in this case, 1370 Hz.), events with frequency in this range are produced by a loop with N = 25. Thus, the four highest frequencies with N = 50 samples (i.e. 1755, 1640, 1549 and 1450 Hz.) are not used. Note, however, that if one were not using timed loops varying by one cycle, corresponding to the n variable (n = 0,1), the frequency spread between possible levels would be twice as great.

The table on the next page gives the actual values produced at the most critical timing regions, as solved from equation (4).

In the case of two-channel sound production, the basic problem is that, instead of just one sample going to the D-A converter at a time, one must send two, usually different values, thus causing a greater minimum delay possible between consecutive samples. An additional 9.0 microseconds are required, five of which are taken by the action of the D-A converter. The timing equation for stereo production with N = 50 samples is:

$$20.4 + 1.6\ T + 0.8\ n = \frac{10^6}{N.F} \qquad \ldots(6) \text{ microseconds}$$

therefore: F_{max} = 980 Hz. for N = 50 samples

Unfortunately, for N = 25 and 17 samples, extra time is needed for incrementing the storage location, thus reducing the maximum frequencies. These cases, then, have equations:

$$22.8 + 1.6\ T + 0.8\ n = \frac{10^6}{N.F} \quad \text{with } F_{max} = 1755 \text{ Hz. for N = 25 samples}$$

$$25.5 + 1.6\ T + 0.8\ n = \frac{10^6}{N.F} \quad \text{with } F_{max} = 2335 \text{ Hz.}$$

for N = 17 samples

Some of the possible frequency values derived from these equations appear in Table IV. By producing frequencies in certain regions by a different value of N, the maximum error between the specified frequency and the result is about ± 2.0 to 2.5%, but only in the critical regions in the faster loops. Below these, for slower loops, the accuracy is much greater, with errors being less than 1%.

TABLE III. Possible Discrete Synthesis Frequencies for Monophonic Output (POD4)

dt (usec.)	T	n	Frequency levels N = 17	Frequency levels N = 25	Frequency levels N = 50
11.4	0	0	5150	3510	
12.2	0	1	4820	3280	not
13.0	1	0	4530	3080	used
13.8	1	1	4260	2900	- - - -
14.6	2	0	4040	2740	1370
15.4	2	1	3820	2600	1300
16.2	3	0	3630	2470	1235
17.0	3	1	- - - -	2350	1175
17.8	4	0	not	2250	1125
18.6	4	1	used	2150	1075
19.4	5	0		2060	1030
20.2	5	1		1980	990
21.0	6	0		1910	955
21.8	6	1		1835	917
22.6	7	0		1770	885
23.4	7	1		1710	855
24.2	8	0		1650	825
25.0	8	1		1600	800
25.8	9	0		1550	775
26.6	9	1		1503	751
27.4	10	0		1460	730
28.2	10	1		1420	710
29.0	11	0		- - - -	690
29.8	11	1		not	670
30.6	12	0		used	654
etc.					etc.

TABLE IV. Possible Discrete Synthesis Frequencies for Two-Channel Output (POD4)

dt (usec)	T	n	Frequency Levels N = 17	N = 25	N = 50
20.4	0	0	not	not	not
21.2	0	1	possible	possible	used
22.0	1	0		- - - -	910
22.8	1	1		1755	878
23.6	2	0		1695	848
24.4	2	1	- - - -	1640	820
25.2	3	0	2335	1590	795
26.0	3	1	2265	1540	770
26.8	4	0	2210	1490	745
27.6	4	1	2155	1450	725
28.4	5	0	2095	1410	705
29.2	5	1	2015	1370	685
30.0	6	0	1960	1335	668
30.8	6	1	1910	1300	650
31.6	7	0	1860	1267	633
32.4	7	1	1805	1235	617
33.2	8	0	1770	1204	602
34.0	8	1	- - - -	1176	588
34.8	9	0	not	1150	575
35.6	9	1	used	1122	561
36.4	10	0		1100	550
37.2	10	1		1075	537
38.0	11	0		1052	526
38.8	11	1		1030	515
39.6	12	0		1010	505
40.4	12	1		990	495
41.2	13	0		970	485
42.0	13	1		952	476
42.8	14	0		935	467
43.6	14	1		916	458
44.4	15	0		- - -	450
etc.				not used	etc.

APPENDIX B: SOUND SYNTHESIS METHODS:

SECTION II: FIXED WAVEFORM SYNTHESIS (POD5):

Some of the problems not solved by the synthesis method used in POD4, as described in Section I, were:

1) The problems in sound quality produced by the decreasing sampling rate, particularly below 600 Hz. The sampling rate is defined as the number of samples per second generated by the computer, and therefore is equal to the number of samples per period, N, multiplied by the number of periods per second, i.e. the frequency F. Therefore the sampling rate F_s may be written:

$$F_s = N \cdot F = 1/dt \qquad \ldots(7)$$

With the variable sampling rate used in these programs, it is necessary to keep increasing the number of samples per period, N, for lower and lower values of F, in order to keep the sampling rate in a reasonable range. With a rate of F_s, the maximum bandwidth that can be generated is $F_s/2$, and for any partial or frequency produced above that limit, a "folded-over" frequency occurs at $F_s - F$. With a low-pass filter, the spectrum above a certain limit may be eliminated, but with too low a sampling rate, folded over frequencies will be produced in the audio region where they cannot be separated from the normal spectrum.

2) In order to achieve smooth envelope control, many discrete amplitude values need to be used, and since it is not possible to calculate and store all of these, amplitude scaling must be done in real time as well. What can be stored, however, are a series of factors by which the stored array can be divided. With the 18 bit word length of the PDP-15, the stored array with values from 0 to 4096 can be shifted 6 places to the left (multiplied by 64), and if 60 amplitude levels are to be used, with 10 equal divisions of 6 powers of 2, the factors will have the form:

$$\text{Factor (I)} = 64. \times 2^{((I-1)/10)} \qquad I = 1,60 \quad ..(8)$$

In order to keep the same zero line for each amplitude level, a constant factor must be added of the form:

$$\text{Constant (I)} = 2048. - \frac{2048.}{\text{Factor(I)}/64.} \qquad \ldots(9)$$

The fastest loop performing all of these operations was found to take 21.0 microseconds per sample. Therefore, we may write a timing equation analogous to equation (4) as:

$$21.0 + 1.6\ T + 0.8\ n = \frac{10^6}{N.F} \quad \ldots(10)$$

If dt is greater than or equal to 21.0×10^{-6} sec., then:

$$F_s \leq 1/(dt) = 47.5 \text{ KHz.}$$

To maintain this sampling rate at F = 100 Hz., then:

$$N \geq F_s/F = 475$$

Therefore an array size of 500 samples was chosen to assure a high sampling rate throughout the range. As well, this size conveniently can be stored on two blocks of dectape. If during the program loop we divide the number of samples in the array by integer values, k, such that:

$$N = 500/k$$

then k should be kept as small as possible for any given frequency in order to maximize N, and thereby the sampling rate. Substituting in equation (10):

$$210 + 16\ T + 8\ n = \frac{2x10^4 \cdot k}{F}$$

for the fastest loop ($T = n = 0$) $\quad k \geq 21 \cdot F/2000$

Choosing k in this way always keeps N high, and as seen in Table V, the range of sampling rates stays remarkably constant, usually above 40K, dipping below 30 K only between 97 and 118 Hz. and below 60 Hz.

A similar method is used for the two channel output, where the maximum sampling rate is about 26KHz. The discrete frequency values for it are listed in Table VI.

Because one continues to cut the number of samples per period in the upper range, there is no strict upper frequency limit. However, with the single channel output, the number of samples per period drops to 5 at 8KHz, and with the two-channel output to 4 at 6KHz; since there is only about another octave in the audible range above these values, this small number is not critical in defining a waveform since higher frequencies need not be generated.

TABLE V: Possible Discrete Synthesis Frequencies for Single Channel Output; Fixed Waveform Synthesis Method, POD5

Array size: 500   Number of samples per period $N = 500/k$

Timing variables: T,n $210 + 16T + 8n = \frac{20000.k}{F}$ sec.

k	N	T	n	F (Hz)	Sampling Rate (KHz)
84	5	0	0	8000	40.0
83	6	0	0	7904	47.4
82		0	0	7809	46.8
81		0	0	7714	46.2
80		0	0	7619	45.7
79		0	0	7523	45.1
78		0	0	7428	44.5
77		0	0	7333	44.0
76		0	0	7238	43.4
75		0	0	7142	42.9
74		0	0	7047	42.3
73		0	0	6952	41.7
72		0	0	6857	41.1
71	7	0	0	6761	47.3
70		0	0	6666	46.7
69		0	0	6571	46.0
68		0	0	6476	45.3
67		0	0	6380	44.7
66		0	0	6285	44.0
65		0	0	6190	43.3
64		0	0	6095	42.7
63		0	0	6000	42.0
62	8	0	0	5904	47.2
61		0	0	5809	46.4
60		0	0	5714	45.7
59		0	0	5619	45.0
58		0	0	5523	44.2
57		0	0	5428	43.4
56		0	0	5333	42.7
55	9	0	0	5238	47.1
54		0	0	5142	46.3
53		0	0	5047	45.4
52		0	0	4952	44.6
51		0	0	4857	43.7
50	10	0	0	4761	47.6
49		0	0	4666	46.7
48		0	0	4571	45.7
47		0	0	4476	44.8
46		0	0	4380	43.8
45	11	0	0	4285	47.1
44		0	0	4190	46.1
43		0	0	4095	45.0
42		0	0	4000	44.0
41	12	0	0	3904	46.8
40		0	0	3809	45.7
39		0	0	3714	44.6
38	13	0	0	3619	47.0
37		0	0	3523	45.8
36		0	0	3428	44.6

TABLE V. Continued:

k	N	T	n	F (Hz)	Sampling Rate (KHz)
35	14	0	0	3333	46.7
34		0	0	3238	45.3
33	15	0	0	3142	47.1
32		0	0	3047	45.7
31	16	0	0	2952	47.2
30		0	0	2857	45.7
29	17	0	0	2761	46.9
28		0	0	2666	45.3
27	18	0	0	2571	46.3
26	19	0	0	2476	47.0
25	20	0	0	2380	47.6
24		0	0	2285	45.7
23	21	0	0	2190	46.0
22	22	0	0	2095	46.1
21	23	0	0	2000	46.0
20	25	0	0	1904	47.6
20		0	1	1834	45.9
19	26	0	0	1809	47.0
19		0	1	1743	45.3
18	27	0	0	1714	46.3
18		0	1	1651	44.6
17	29	0	0	1619	47.0
17		0	1	1559	45.2
16	31	0	0	1523	47.2
16		0	1	1467	45.5
15	33	0	0	1428	47.1
15		0	1	1376	45.4
14	35	0	0	1333	46.6
14		0	1	1284	44.9
13	38	0	0	1238	47.0
13		0	1	1192	45.3
12	41	0	0	1142	46.8
12		0	1	1100	45.1
11	45	0	0	1047	47.1
11		0	1	1009	45.4
11		1	0	973	43.8
10	50	0	0	952	47.6
10		0	1	917	45.9
10		1	0	884	44.2
9	55	0	0	857	47.1
9		0	1	825	45.4
9		1	0	796	43.8
8	62	0	0	761	47.2
8		0	1	733	45.4
8		1	0	707	43.8
8		1	1	683	42.3
7	71	0	0	666	47.3
7		0	1	642	45.6
7		1	0	619	44.0
7		1	1	598	42.5
6	83	0	0	571	47.4
6		0	1	550	45.7
6		1	0	530	44.0
6		1	1	512	42.5
6		2	0	495	41.1

TABLE V: Continued:

k	N	T	n	F (Hz)	Sampling Rate (KHz.)
6	83	2	1	489	39.8
5	100	0	0	476	47.6
5		0	1	458	45.8
5		1	0	442	44.2
5		1	1	427	42.7
5		2	0	413	41.3
5		2	1	400	40.0
5		3	0	387	38.7
4	125	0	0	380	47.5
4		0	1	366	45.7
4		1	0	353	44.1
4		1	1	341	42.6
4		2	0	330	41.3
4		2	1	320	40.0
4		3	0	310	38.8
4		3	1	300	37.5
4		4	0	291	36.4
3	166	0	0	285	47.3
3		0	1	275	45.7
3		1	0	265	44.0
3		1	1	256	42.5
3		2	0	247	41.0
3		2	1	240	39.8
3		3	0	232	38.5
3		3	1	225	37.4
3		4	0	218	36.2
3		4	1	212	35.2
3		5	0	206	34.2
3		5	1	201	33.4
3		6	0	196	32.5
3		6	1	191	31.7
2	250	0	0	190	47.5
2		0	1	183	45.8
2		1	0	176	44.0
2		1	1	170	42.5
2		2	0	165	41.3
2		2	1	160	40.0
2		3	0	155	38.8
2		3	1	150	37.5
2		4	0	145	36.3
2		4	1	141	35.3
2		5	0	137	34.3
2		5	1	134	33.5
2		6	0	130	32.5
2		6	1	127	31.8
2		7	0	124	31.0
2		7	1	121	30.3
2		8	0	118	29.5
2		8	1	115	28.8
2		9	0	112	28.0
2		9	1	110	27.5
2		10	0	108	27.0
2		10	1	105	26.3

TABLE V: Continued:

k	N	T	n	F (Hz)	Sampling Rate (KHz.)
2	250	11	0	103	25.8
2		11	1	101	25.2
2		12	0	99	24.8
2		12	1	97	24.3
1	500	0	0	95	47.5
1		0	1	91	45.5
1		1	0	88	44.0
1		1	1	85	42.5
1		2	0	82	41.0
1		2	1	80	40.0
1		3	0	77	38.5
1		3	1	75	37.5
1		4	0	72	36.0
1		4	1	70	35.0
1		5	0	68	34.0
1		5	1	67	33.5
1		6	0	65	32.5
1		6	1	63	31.5
1		7	0	62	31.0
1		7	1	60	30.0
1		8	0	59	29.5
1		8	1	57	28.5
1		9	0	56	28.0
1		9	1	55	27.5
1		10	0	54	27.0
1		10	1	52	26.0
1		11	0	51	25.5
1		11	1	50	25.0

TABLE VI. Possible Discrete Synthesis Frequencies for Two-Channel Output; Fixed Waveform Synthesis Method: POD5

Array size: 500 Number of samples per period N = 500/k

Timing variables T,n

$$373 + 16T + 8n = \frac{20000.k}{F} \text{ sec.}$$

k	N	T	n	F (Hz)	Sampling Rate (KHz)
112	4	0	0	6005	24.0
111		0	0	5951	23.8
110		0	0	5898	23.6
109		0	0	5844	23.4
108		0	0	5790	23.1
107		0	0	5737	22.9
106		0	0	5683	22.7
105		0	0	5630	22.5
104		0	0	5576	22.3
103		0	0	5522	22.1
102		0	0	5469	21.9
101		0	0	5415	21.7
100	5	0	0	5361	26.8
99		0	0	5308	26.5
98		0	0	5254	26.3
97		0	0	5201	26.0
96		0	0	5147	25.7
95		0	0	5093	25.4
94		0	0	5040	25.2
93		0	0	4986	24.9
92		0	0	4932	24.7
91		0	0	4879	24.4
90		0	0	4825	24.1
89		0	0	4772	23.9
88		0	0	4718	23.6
87		0	0	4664	23.3
86		0	0	4611	23.1
85		0	0	4557	22.8
84		0	0	4504	22.5
83	6	0	0	4450	26.7
82		0	0	4396	26.4
81		0	0	4343	26.1
80		0	0	4289	25.7
79		0	0	4235	25.4
78		0	0	4182	25.1
77		0	0	4128	24.8
76		0	0	4075	24.5
75		0	0	4021	24.1
74		0	0	3967	23.8
73		0	0	3914	23.5
72		0	0	3860	23.2
71	7	0	0	3806	26.6
70		0	0	3753	26.3
69		0	0	3699	25.9
68		0	0	3646	25.5
67		0	0	3592	25.1
66		0	0	3538	24.8

TABLE VI. Continued:

k	N	T	n	F (Hz)	Sampling Rate (KHz)
65	7	0	0	3485	24.4
64		0	0	3431	24.0
63		0	0	3378	23.6
62	8	0	0	3324	26.6
61		0	0	3270	26.2
60		0	0	3217	25.7
59		0	0	3163	25.3
58		0	0	3109	24.9
57		0	0	3056	24.4
56		0	0	3002	24.0
55	9	0	0	2949	26.5
54		0	0	2895	26.1
53		0	0	2841	25.6
52		0	0	2788	25.1
51		0	0	2734	24.6
50	10	0	0	2680	26.8
49		0	0	2627	26.3
48		0	0	2573	25.7
47		0	0	2520	25.2
46		0	0	2466	24.7
45	11	0	0	2412	26.5
44		0	0	2359	25.9
43		0	0	2305	25.4
42		0	0	2252	24.8
41	12	0	0	2198	26.4
40		0	0	2144	25.7
39		0	0	2091	25.1
38	13	0	0	2037	26.5
37		0	0	1983	25.8
36		0	0	1930	25.1
35	14	0	0	1876	26.3
34		0	0	1823	25.5
33	15	0	0	1769	26.5
32		0	0	1715	25.7
32		0	1	1679	25.2
31	16	0	0	1662	26.6
31		0	1	1627	26.0
30		0	0	1608	25.7
30		0	1	1574	25.2
29	17	0	0	1554	26.4
29		0	1	1522	25.9
28		0	0	1501	25.5
28		0	1	1469	25.0
27	18	0	0	1447	26.0
27		0	1	1417	25.5
26	19	0	0	1394	26.5
26		0	1	1364	25.9
25	20	0	0	1340	26.8
25		0	1	1312	26.2
24		0	0	1286	25.7
24		0	1	1259	25.2
23	21	0	0	1233	25.9
23		0	1	1207	25.3

TABLE VI. Continued:

k	N	T	n	F (Hz)	Sampling Rate (KHz)
22	22	0	0	1179	25.9
22		0	1	1154	25.4
21	23	0	0	1126	25.9
21		0	1	1102	25.3
20	25	0	0	1072	26.8
20		0	1	1049	26.2
20		1	0	1028	25.7
19	26	0	0	1018	26.5
19		0	1	997	25.9
19		1	0	976	25.4
18	27	0	0	965	26.1
18		0	1	944	25.5
18		1	0	925	25.0
17	29	0	0	911	26.4
17		0	1	892	25.9
17		1	0	874	25.3
16	31	0	0	857	26.6
16		0	1	839	26.0
16		1	0	822	25.5
15	33	0	0	804	26.5
15		0	1	787	26.0
15		1	0	771	25.4
14	35	0	0	750	26.3
14		0	1	734	25.7
14		1	0	719	25.2
13	38	0	0	697	26.5
13		0	1	682	25.9
13		1	0	668	25.4
13		1	1	654	24.9
12	41	0	0	643	26.4
12		0	1	629	25.8
12		1	0	616	25.3
12		1	1	604	24.8
11	45	0	0	589	26.1
11		0	1	577	26.0
11		1	0	565	25.4
11		1	1	554	24.9
11		2	0	543	24.4
10	50	0	0	536	26.8
10		0	1	524	26.2
10		1	0	514	25.7
10		1	1	503	25.2
10		2	0	493	24.7
9	55	0	0	482	26.5
9		0	1	472	26.0
9		1	0	462	25.4
9		1	1	453	24.9
9		2	0	444	24.4
9		2	1	435	23.9
8	62	0	0	428	26.5
8		0	1	419	26.0
8		1	0	411	25.5
8		1	1	403	25.0

TABLE VI. Continued:

k	N	T	n	F (Hz)	Sampling Rate (KHz)
8	62	2	0	395	24.5
8		2	1	387	24.0
8		3	0	380	23.6
7	71	0	0	375	26.6
7		0	1	367	26.1
7		1	0	359	25.5
7		1	1	352	25.0
7		2	0	345	24.5
7		2	1	338	24.0
7		3	0	332	23.6
7		3	1	326	23.1
6	83	0	0	321	26.6
6		0	1	314	26.1
6		1	0	308	25.6
6		1	1	302	25.1
6		2	0	296	24.6
6		2	1	290	24.1
6		3	0	285	23.7
6		3	1	279	23.2
6		4	0	274	22.7
5	100	0	0	268	26.8
5		0	1	262	26.2
5		1	0	257	25.7
5		1	1	251	25.1
5		2	0	246	24.6
5		2	1	242	24.2
5		3	0	237	23.7
5		3	1	233	23.3
5		4	0	228	22.8
5		4	1	224	22.4
5		5	0	220	22.0
4	125	0	0	214	26.8
4		0	1	209	26.1
4		1	0	205	25.6
4		1	1	201	25.1
4		2	0	197	24.6
4		2	1	193	24.1
4		3	0	190	23.8
4		3	1	186	23.3
4		4	0	183	22.9
4		4	1	179	22.4
4		5	0	176	22.0
4		5	1	173	21.6
4		6	0	170	21.3
4		6	1	167	20.9
4		7	0	164	20.5
4		7	1	162	20.3
3	166	0	0	160	26.6
3		0	1	157	26.1
3		1	0	154	25.6
3		1	1	151	25.1
3		2	0	148	24.6
3		2	1	145	24.1

TABLE VI. Continued:

k	N	T	n	F (Hz)	Sampling Rate (KHz)
3	166	3	0	142	23.6
3		3	1	139	23.1
3		4	0	137	22.7
3		4	1	134	22.2
3		5	0	132	21.9
3		5	1	130	21.6
3		6	0	127	21.1
3		6	1	125	20.8
3		7	0	123	20.4
3		7	1	121	20.0
3		8	0	119	19.8
3		8	1	117	19.4
3		9	0	116	19.3
3		9	1	114	18.9
3		10	0	112	18.6
3		10	1	110	18.3
3		11	0	109	18.1
2	250	0	0	107	26.8
2		0	1	104	26.0
2		1	0	102	25.5
2		1	1	100	25.0
2		2	0	98	24.5
2		2	1	96	24.0
2		3	0	95	23.5
2		3	1	93	23.3
2		4	0	91	22.8
2		4	1	89	22.3
2		5	0	88	22.0
2		5	1	86	21.5
2		6	0	85	21.3
2		6	1	83	20.8
2		7	0	82	20.5
2		7	1	81	20.3
2		8	0	79	19.8
2		8	1	78	19.5
2		9	0	77	19.3
2		9	1	76	19.0
2		10	0	75	18.8
2		10	1	73	18.3
2		11	0	72	18.0
2		11	1	71	17.8
2		12	0	70	17.5
2		12	1	69	17.3
2		13	0	68	17.0
2		13	1	67	16.8
2		14	1	66	16.5
2		15	0	65	16.3
2		15	1	64	16.0
2		16	0	63	15.8
2		16	1	62	15.5
2		17	1	61	15.3
2		18	0	60	15.0
2		18	1	59	14.8
2		19	1	58	14.5

APPENDIX B: SOUND SYNTHESIS METHODS:

SECTION III: FREQUENCY MODULATION METHOD: (POD6):

The equation describing an acoustic wave produced by a carrier sine wave, frequency c, modulated by another sine wave, frequency m so that the frequency of the carrier wave goes between c+d and c−d, where d is the deviation of the modulating wave (i.e. its amplitude), may be written as:

$$e = A \cdot \sin(2\pi ct + I \cdot \sin(2\pi mt)) \qquad \ldots(11)$$

where I is the modulation index such that $I = d/m$

If we have stored a sine array of the form:

$$E_J = A \cdot \sin\left(\frac{2\pi J}{N}\right) + 2048. \qquad J = 1,\ldots,N \qquad \ldots(12)$$

then we may construe the problem of producing the function described in (11) as the problem of finding the index J of the next sample to be put out. Therefore, equating equations (11) and (12), ignoring amplitudes for the moment, and comparing arguments of the sine function, we have:

$$J = N \cdot \left(ct + \frac{I}{2\pi} . \sin 2\pi mt\right) \qquad \ldots(13)$$

In order to deal with the time variable, we can make an initial assumption that the modulating frequency will be periodic, and divided into i segments of dt. Therefore:

$t = n \cdot dt$ such that $T_m/i = dt$ where T_m is the period of modulation

Therefore: $t = \dfrac{n}{m.i}$, $n = 0,1,2,\ldots,i$

Substituting in (13): $$J = N \cdot \left(\frac{c.n}{m.i} + \frac{I}{2\pi} . \sin 2\pi \frac{n}{i}\right) \qquad \ldots(14)$$

Note that the sine term on the right now is independent of t, and therefore can be compared with the stored array which has the specific form:

$$E_M = 2048. + 2048. \sin \frac{2\pi M}{N} \quad , \quad M = 0,1,2,\ldots,N$$

If $i \leqslant N$, such that $N/i = k$, then $E_M = E_{k.n}$

Therefore: $\sin 2\pi \frac{n}{i} = (E_{k.n} - 2048)/2048$

It will also be to our advantage to scale I over a larger range, which we can do at the same time as removing the factor π by defining:

$$I' = 16.I/\pi$$

Substituting in (14):

$$J = k.(\frac{c}{m}).n + \frac{N.I'}{16}.\frac{(E_{k.n}-2048)}{4096} \quad ...(15)$$

Equation (15) is now programmable in terms of the variables c,m and I for a given array size N, with the subdivisions of the modulating period being counted by n, n = 1,2,...,i. In the present case, N = 512, then equation (15) can rewritten as:

$$J = k.(\frac{c}{m}).n + \frac{I'}{256}.(E_{k.n}-2048)$$

where $k = N/i = 512/i$

However, much of this equation includes constants which are multiplied by an expression dependent on the various values of n. There will then be only i different values of these for each event, and therefore they can be calculated beforehand and stored in two small arrays A_n and B_n :

$$A_n = k.(\frac{c}{m}).n \quad \text{and} \quad B_n = (E_{k.n}-2048)/8$$

Therefore, the equation finally reduces to the form:

$$J = A_n + I'.B_n/16 \quad ...(16)$$

The factor 16 is retained so that the multiplication in the numerator can be carried out over the widest possible range for accuracy. The values calculated from this expression have to be adjusted to the array size with a modulus function if they exceed 512, or by adding the array size to negative values till they return to the positive range. As well, the last index in the modulation period must be carried over as a phase constant for the next period in order to ensure a continuity in the carrier wave. The initial value of this phase constant determines the relationship between the carrier and modulating wave, and in the present program is initialized to 128, that is, a cosine modulation.

The speed of the program loop performing these operations is 58.8 microseconds; thus the timing equation can be written:

$$588 + 16\,T + 8\,n = \frac{10^7}{m.i} \quad ...(17)$$

where T and n are the timing variables as used in sections I, II of this discussion. Again, the number of samples per period (i) may vary in order to keep the sampling rate as high as possible, such that $i = 512/k$.

The difference in the case of this program is that k may only have values that produce discrete values of i. However, substituting in equation (17),

$$588 + 16\ T + 8\ n = \frac{19531.k}{m}$$

Therefore: $$k \geqslant \frac{588.m}{19531}$$

and: $$T = \frac{19531/(m/k) - 588}{16}$$

The discrete frequencies produced are similar to those of Tables V & VI., with sampling rates in the region of 13 to 16K. over most of the modulation frequency range. The fastest loops (T = n = 0) for some values of i are listed below, with the slowest one for the same i on the right.

Size of array: 512. Number of samples per period i = 512/k

k	i	F (Hz)	Sampling rate (KHz)	F (Hz)	Sampling Rate (KHz)
128	4	4251	17.0	3415	13.7
102	5	3388	16.9	2845	14.2
85	6	2823	16.9	2455	14.7
73	7	2424	17.0	2160	15.1
64	8	2125	17.0	1871	15.0
56	9	1860	16.8	1719	15.5
51	10	1694	16.9	1546	15.5
46	11	1527	16.8	1412	15.5
42	12	1395	16.7	1306	15.7
39	13	1295	16.8	1212	15.8
36	14	1195	16.7	1134	15.9
34	15	1129	16.9	1071	16.1
32	16	1062	17.0	1008	16.1
30	17	996	16.9	945	16.1
28	18	930	16.7	870	15.7
26	19	863	16.4	840	16.0
25	20	830	16.6	808	16.2
24	21	797	16.7	776	16.3
23	22	763	16.8	734	16.1
22	23	730	16.8	702	16.1
21	24	697	16.7	670	16.1
20	25	664	16.6	638	16.0
19	26	631	16.4	606	15.8
18	28	597	16.7	567	15.9
17	30	564	16.9	535	16.1
16	32	531	17.0	504	16.1
15	34	498	16.9	472	16.0
14	36	465	16.7	435	15.7
13	39	431	16.8	404	15.8
12	42	398	16.8	368	15.4
11	46	365	16.8	337	15.5
10	51	332	16.9	303	15.4
9	56	298	16.7	269	15.1

k	i	F (Hz)	Sampling Rate (KHz)	F (Hz)	Sampling Rate (KHz)
8	64	265	17.0	236	15.1
7	73	232	16.9	202	14.7
6	85	199	16.9	167	14.2
5	102	166	16.9	133	13.5
4	128	132	16.9	100	12.8
3	170	99	16.8	69	11.7
2	256	66	16.9	34	8.7
1	512	33	16.9		

TABLE VII. Summary of discrete modulation frequency levels (fastest and slowest loops for given i = no. of samples/period) POD6 synthesis method.

TABLE VIII. Some Bessel Functions giving amplitude of frequency modulation sidebands for various modulation indices (I):

I	n = 0	1	2	3	4	5	6	7	8	9	10	11	12
0.05	1.00	.025											
0.1	1.00	.05											
0.2	.99	.10											
0.3	.98	.15											
0.4	.96	.20											
0.5	.94	.24	.03										
0.6	.91	.29	.04										
0.8	.85	.37	.07	.01									
1.0	.77	.44	.11	.02									
1.2	.67	.50	.17	.03									
1.4	.57	.54	.21	.05	.01								
1.6	.46	.57	.26	.07	.02								
2.0	.22	.58	.35	.13	.03								
2.4	.003	.52	.43	.20	.06								
3.0	.26	.34	.49	.31	.13	.04	.01						
4.0	.40	.07	.36	.43	.28	.13	.05	.02					
5.0	.18	.33	.05	.36	.39	.26	.13	.05	.02				
6.0	.15	.28	.24	.11	.36	.36	.25	.13	.06	.02			
7.0	.30	.005	.30	.17	.16	.35	.34	.23	.13	.06	.01	.01	
8.0	.17	.23	.11	.29	.11	.19	.34	.32	.22	.13	.06	.03	.01
9.0	.09	.25	.14	.18	.27	.06	.20	.33	.31	.21	.12	.06	.03
10.0	.25	.04	.25	.06	.22	.23	.01	.22	.32	.29	.21	.12	.06
12.0	.05	.22	.09	.20	.18	.07	.24	.17	.05	.23	.30	.27	.20
15.0	.01	.21	.04	.19	.12	.13	.21	.03	.17	.22	.09	.10	.24
18.0	.01	.19	.01	.19	.07	.16	.16	.05	.20	.12	.07	.20	.18
21.0	.04	.17	.02	.18	.03	.16	.11	.10	.18	.03	.15	.17	.03
24.0	.06	.15	.04	.16	.003	.16	.06	.13	.14	.04	.17	.10	.07
	n = 0	1	2	3	4	5	6	7	8	9	10	11	12

APPENDIX C: CHOICE CODE FOR THE CONTROL PROGRAM "SAM":

SECTION I: POD4 & 5:

P : Performance and synthesis:

- P1 - Play the distribution of events with test waveform
- P2 - Play the distribution of events with waveform selection and stereo option
- P3 - Waveform test with frequency, envelope variables (monophonic)
- P4 - Play all compositional sections through from the beginning
- P5 - Two-channel waveform test with frequency, envelope and spatial position variables

W : Waveform Selection: (POD4); Object Selection: (POD5):

- W1 - Waveform specification and selection
- W2 - Change waveform numbers
- W3 - Change waveform tendency mask (single data value)
- W4 - Change selection ratios
- W5 - Change waveform selection principle

D : Distribution Characteristics:

- D1 - Information concerning theoretical distribution characteristics
- D2 - Analysis of actual distribution as calculated (densities and average time delays) taking into account all performance variables affecting the time structure)
- D3 - Change frequency scale to logarithmic for individual mask segments
- D4 - Change density to theoretically equal values throughout distribution
- D5 - Specify theoretical distribution with initial and final densities

R : (Re)-Calculation of the Poisson Distribution

- R1 - (Re)-Calculate distribution and play with test waveform (P1)
- R2 - (Re)-Calculate distribution and play with waveform selection (P2)
- R3 - (Re)-Calculate distribution without performance
- R4 - Begin a new compositional section (and punch performance data if desired)
- R5 - Restart current section

V : Performance Variables (no recalculation necessary)

- V1 - Change speed of performance (faster below 300/slower above 300)
- V2 - Specify or change envelope values
- V3 - Make entry delay independent of duration
- V4 - Make entry delay dependent on duration (as in initial case)
- V5 - Specify amplitude modulation speed (POD5 only)

S : Spatial Distribution:

S1 - Aleatoric spatial distribution
S2 - Frequency correlation with spatial position (position numbers for each octave: linear/ordered)
S3 - Time correlation with spatial position (tendency mask)
S4 - Change tendency mask data (single value)
S5 - Return to monophonic output (i.e. negate S1, S2 or S3)

F : Waveform Input and Operations:

F1 - Calculate a waveform from the formula library and store on dectape
F2 - Input from paper tape data
F3 - Hybrid waveform construction from data on dectape
F4 - Transfer waveform data to teletype, paper tape or another block on dectape
F5 - Waveform construction with harmonic structure

C : Change data for Poisson distribution:

C1 - Change total number of events
C2 - Change initial density (sounds/sec.)
C3 - Change tendency mask data (single value)
C4 - Insert a new tendency mask segment, or add an additional one
C5 - Change random start number (i.e. for a new variant)

L : List data on teletype:

L1 - List total number of events in the theoretical distribution
L2 - List Poisson frequency tendency mask data
L3 - List waveform tendency mask data
L4 - List spatial distribution tendency mask data
L5 - List envelope data (for all waveforms being used)
List object data (POD5)

T : Data Tapes:

T1 - Punch new Poisson distribution data tape using current values
T2 - Punch waveform distribution tendency mask data tape
T3 - Punch spatial distribution tendency mask data tape
T4 - Punch co-ordinates of calculated frequency-time points
T5 - Punch envelope data tape (POD4); object data tape (POD5)

G : Graphic Display:

G1 - Make a graph of the Poisson distribution co-ordinates
G2 - Make a graph of the Poisson tendency mask
G3 - Make a graph of the waveform tendency mask
G4 - Make a graph of the spatial tendency mask
G5 - Make a graph of waveform data (five sections)

H : Halt the program

APPENDIX C: CHOICE CODE FOR THE CONTROL PROGRAM:

SECTION II: POD6:

P : Performance and Synthesis:

P1 - Play the distribution of events with a test waveform
P2 - Play the distribution of events with object selection
P3 - Single sound test with envelope. c-m ratio and maximum modulation index as variables
P4 - Play all compositional sections through from the beginning
P5 - Repeated sound test (similar to P3) terminated by setting AC switch

W : Object Selection:

W1 - Object specification of availability for a distribution and stipulation of selection principle
W2 - Change object numbers to be used
W3 - Change object selection tendency mask (single value)
W4 - Change selection ratios
W5 - Change object selection principle

Note: to add a tendency mask section ask for W5 indicating a larger number of mask sections. The program then gives the option of adding new sections without retyping the entire mask

Note: to add new objects to a tendency mask without changing the selection of the previously used objects, ask for W1 or W2 stating a larger number of object; the program then offers to scale the old mask so that it still only applies to the original objects. One may then add new mask segments as in the previous note.

D : Distribution Characteristics:

D1 - Information concerning the theoretical distribution characteristics
D2 - Analysis of actual distribution as calculated and performed, giving sound densities, average time delays, taking into account all performance variables such as clock setting and performance speed
D3 - Change frequency scale of individual mask segments to logarithmic
D4 - Change density to theoretically equal values throughout the distribution (ask for L1 afterwards to learn the new number of events)
D5 - Specify theoretical distribution densities with initial and final values (real numbers)- ask for L1 after to learn new total number of events

R : (Re)-Calculation of the Poisson Distribution

R1 - (Re)-calculate distribution and play with test waveform (P1)
R2 - (Re)-calculate distribution and play with object selection (P2)
R3 - (Re)-calculate distribution without performance
R4 - Begin new compositional section, produce performance tape (optional) and list all data
R5 - Restart current section (Poisson data asked for next time but other data preserved)

V : Performance Variables (no recalculation necessary)

V1 - Change speed of performance (initially 300; faster below 300, slower above 300)
V2 - Specify object data from teletype (only). Asks for object number, envelope, c-m ratios and maximum modulation index the first time. Thereafter, used to change envelope only.
V3 - Make entry delay independent of duration
V4 - Make entry delay dependent on duration (as in initial case)
V5 - Change c-m ratio and modulation index of given object

C : Change data for Poisson distribution

C1 - Change total number of events
C2 - Change initial density (sounds/sec.)
C3 - Change tendency mask value (single value)
C4 - Insert a new tendency mask segment, or add an additional one
C5 - Change random start number (initially 1)

L : List data on teletype

L1 - List total number of events in the theoretical distribution
L2 - List Poisson frequency tendency mask data
L3 - List object selection tendency mask data
L4 -
L5 - List object data (for all object defined)

T : Punch data tapes

T1 - Punch new Poisson distribution data tape using current values
T2 - Punch object selection tendency mask data tape
T3
T4 - Punch co-ordinates of calculated frequency-time points
T5 - Punch object data tape (all objects defined)

Q : Object Information

O1 - Object information regarding a specific c-m ratio: gives spectrum and optimum synthesis information
O2 - Object information regarding the user's set of available objects (used only after W1 and object definition)
O3 - Input of object data tape

G : Graphic Display

G1 - Make a graph of the Poisson distribution co-ordinates (points)
G2 - Make a graph of the Poisson tendency mask
G3 - Make a graph of the waveform tendency mask

H : Halt the program

APPENDIX D: THE OVERLAY STRUCTURE OF THE PROGRAM:

Allocation of 12K core for programs, data storage, input-output, handlers.

Link Table
Main routine: subroutine calls, etc.
¨ Monitor: coding of commands: subroutine calls
¨ Integer input
Random generator
Fortran arithmetic subroutines
Permanent data storage (Common arrays)

(4K)

¨ Major Synthesis Subroutine ¨ Waveform data Block (2500) (4K)	Selection Restart prog. Section data ¨ Test synthesis ¨ Vocabulary	Poisson input and calcul'n. Tendency calcul'n. Area calc'n. Vocabulary Waveform fitting	Waveform handlers (hybrid, harmonic, transfer ¨ Graphic routine	Distrib'n. inform'n. & analysis BCDIO Waveform formulas

---

¨ Dectape handler (integer input and output)	¨ Vocabulary	¨ Integer input and output (teletype, paper punch)
¨ Data block for Synthesis (504)	¨ Distribution Analysis data block	Object Specification Input/Output

(05734_8)

System handlers (dectape, teletype, paper reader & punch)

(4K)

¨ Machine language routines (MACRO-15); others in FORTRAN IV