Y. Höller. *Arcus.* Erato compact disc 2292-45509-2 (with Dufourt, Ferneyhough, Harvey).
S. Haynes. *Prisms.* Wergo compact disc 2026-2 (Computer Music Currents n° 8, with works by Boesch, Rai, eitelbaum, Teruggi).
J.C. Risset. *Computer Suite from Little Boy* (1968). Wergo compact disc 2013-50 (with *Songes , Passages, Sud*).
J.C. Risset. *Mutations* (1969). INA compact disc C1003 (with *Dialogues, Inharmonique, Sud*).
J.C. Risset. *Moments newtoniens* (1977). Mille et un poèmes, vol. 1, compact disc Radio-France, K 15001 AD 100.
J.C. Risset. *Contours* (1983). Neuma compact disc 450-71 (New music series vol. 1).
J.C. Risset. *L'autre face* (1983). Wergo compact disc 2025-2 (Computer Music Currents n° 7, with works by Bodin, Karpen, Petersen, Yuasa).
J.C. Risset. *Voilements* (1987). INA compact disc C2000 Sax-Computer, by Daniel Kientzy (with works by Racot and Teruggi).

AN INTRODUCTORY CATALOGUE OF COMPUTER SYNTHESIZED SOUNDS

---

by J. C. Risset
Bell Telephone Laboratories
Murray Hill, New Jersey, 1969

ACKNOWLEDGMENT

I dedicate this catalogue to Max V. Mathews as a token of admiration and gratitude. It was indeed a great fortune and a great pleasure for me to work with him and to use the wonderful new means he forged to make music.

ABSTRACT

This introductory catalogue presents some 25 examples of sounds generated by computer, using M. V. Mathews' Music V programs. Some of the sounds are instrument-like; some are not. The catalogue consists of the combination of a tape (or a record) of the sounds, which permits one to evaluate them aurally, and of the computer data used for the synthesis of the sounds, which affords a thorough description of the physical structure of these sounds. This is intended as an example to be followed by people working in sound synthesis, so that others can benefit from their findings and so that an extended repertory of sounds can be made available for tone quality studies and for computer music.

Table of Contents

Introductory Notes

This limited "catalogue" presents examples of various types of musical sounds generated by computer, using MUSIC V programs. A general description of the synthesis process is given in reference (1); more details on both the process and the particular program used can be found in reference (2). For each synthesis the user of the programs must provide data which corresponds to the physical parameters of the desired sound. The data used to synthesize a musical excerpt will be from now on referred to as the computer "score" for that excerpt.

It has long been recognized (3) that in order to take advantage of the unlimited resources of computer synthesis of sound, one had to develop a body of psychoacoustical knowledge, enabling him to specify the physical parameters corresponding to a desired type of sound. Experiments with the seemingly well-known sounds of some musical instruments (4) have shown that such knowledge was still very poor, but that computer synthesis of sound was an invaluable tool to remedy this situation.

This catalogue presents results of computer syntheses for some instrument-like and some non-instrument-like sounds. It consists of the combination of a tape (or a record) of the sounds, which permits the evaluation of these sounds aurally, and of the corresponding computer scores with some additional explanations, which gives the recipe for synthesis and also affords a thorough description of the physical structure of the sounds. Thus the reader-listener can relate the physical parameters of the sounds and their subjective effect. He is also able to resynthesize the sounds by using the same or other programs, or any process enabling him to control the necessary physical parameters.

Each example presented is numbered on the tape and on the write-up. Together with the score, some explanations are given on the

purpose of the example, on the design of the instrument, and on the stored functions used. Ahead of the examples a description and a listing are provided for a CONVT subroutine and some GEN subroutines used in the examples but not described in the MUSIC V manual (2).

It must be emphasized that the sounds are presented as examples and by no means as models. In several instances no attempt has been made to optimize the synthesis with regard to simplicity or efficiency; also most of the instrument-like sounds do not attempt a close imitation of real sounds. In our experience, examples of certain types of sounds with their description are most useful, since this provides a starting point for a systematic exploration of the synthesis of sounds of these types: it is then rather straightforward to find and discard unimportant features by systematic variations of the parameters. For the sounds presented here, the physical parameters have been deduced from data on real musical instruments or from the results of various synthesis attempts.

Several of the syntheses presented are not very economical. Simple and economical syntheses are in general easy to explore, and complexity seems often necessary to generate varied sounds with life and musical interest. Yet there exist economical and non-trivial ways to synthesize interesting sounds: for instance, through the use of unusual frequency modulations, as explored by John Chowning at Stanford University; or through the use of non-linear transfer of waves or ring-modulation-like operations, as exemplified in #150 and #550 of this catalogue.

Some of the sound examples presented (#490, 502, 503, 512, 517) are not directly output from the computer but obtained from mixing one or several computer runs. Obviously mixing deprives the user of some of the computer's precision and convenience, and it requires good electroacoustic equipment. Yet, as discussed in #512, it helps the user to control the balance of amplitudes of several voices, and it may permit the same computer runs to be used repeatedly.

Listeners are encouraged to listen to the examples at different tape speeds or backwards: these easy manipulations correspond to simple changes in the physical parameters.

The numbers of the examples are in general nonconsecutive; this is to permit us to insert later new examples at what seems the most logical place. Yet it must be noted that no attempt has been made to classify the sounds presented in a rigorous way. The problems here are formidable, since the dimensionality of timbre perception seems quite high.

This catalogue is only a by-product of some sound explorations, but we hope that it will stimulate other people working in the field of synthetic sound to do the same kind of presentation of their work: then one could take advantage of their results, and an extended repertory of sounds would gradually build up and be made readily available, which could benefit studies in tone quality and perhaps other fields of psycho-acoustics (5) as well as computer music.

References

(1) M. V. Mathews. "The Digital Computer as a Musical Instrument" *Science, 142* (1963) pp.553-557.

(2) M. V. Mathews. *The Technology of Computer Music.* M.I.T. Press, Cambridge, Mass., 1969.

(3) M. V. Mathews, J. R. Pierce, & N. Guttman. "Musical Sounds from Digital Computers", *Gravesaner Blätter* 23/24 (1962) p. 109.

(4) J. C. Risset & M. V. Mathews. "Analysis of Instrument Tones", *Physics Today, 22*, No. 2 (Feb. 1969) pp.23-30.

(5) R. N. Shepard. "Circularity in Judgments of Relative Pitch", *J.Acoust.Soc.Am., 36* (1964) pp.2346-2353.

(6) J. C. Risset. "Pitch Control and Pitch Paradoxes Demonstrated with Computer-Synthesized Sound", *J.Acoust. Soc.Am., 46* (Pt.I) (1969) p.88 (abstract only).

(7) M. V. Mathews. "The Computer Music Record Supplement", *Grav. Blatter* 26 (1965) p. 117.

(8) J. C. Tenney. "The Physical Correlates of Timbre," *Grav. Blatter* 26 (1965) pp.106-109.

(9) J. R. Pierce, M. V. Mathews, & J. C. Risset. "Further Experiments on the Use of the Computer in Connection with Music", *Grav. Blatter* 27/28 (1965) pp.92-97.

APPENDIX

General CONVT Subroutine

For several of the runs which follow, a "general CONVT" has been used. It has been designed by P. Ruiz to perform standard conversions without having to change the subroutine. One specifies for each instrument (from #1 to #5) which P field must undergo which conversion by setting Pass II variables in the following way:

for instrument #1: SV2 0 10 i N_1 N_2 . . . N_i ;

#2: SV2 0 20 . . .

i is the number of note cards fields to be converted.

If one wants to convert P6 as a frequency (that is, P(6) = (Function length/Sampling rate) $*$P(6)) one sets $N_1 = 6$.

If one wants to convert P7 as a duration increment (that is, P(7)=(Function length/Sampling rate)/P(6)), one sets $N_2 = -7$.

The conversion to increments for the ENVelope generator is done as follows. One must provide 3 fields, the 1st one (e.g., P8) for attack time (in s), the 2nd one (e.g., P9) for steady state time, the 3rd one (e.g., P10) for decay time. On the note card P9 is dummy, only attack and decay times need to be specified to their actual value. The CONVT will determine the steady state time by subtracting

P8+P10

from P4 (duration of the note)(if the result is negative it will assume steady state duration 0 (e.g., P9=Function length/4) and shorten P8 and P10 so that P8+P10=P4).

It will then apply conversion $P_{(j)}$=(Function length/4$*$Sampling rate)/$P_{(j)}$). To get P8, P9, P10 converted this way, one simply sets N_4 =108. This CONVT provides also for conversion for the FLT (filter) unit generator (not used in the examples).


```
CGCONVT                    GENERAL CONVT
C* * * * CONVERT* * * FREQUENCY* * TIME* * ENV. PARAMETERS
C* * * * FOR INSTR 1 SV2 0 10 N N1 N2 N3 . . .
C* * * * FOR INSTR 2 SV2 0 20 . . .
C* * * * N NUMBER OF FIELDS TO CONVERT
C* * * * N1, N2 . . . FIELDS NUMBERS
C* * * * * * * * * * * * *
C* * * * E.G. TO CONVT P6 AS A FREQUENCY N1=6
C* * * TO CONVT P7 AS A TIME INCREMENT N2=-7
C* * *TO HAVE P8 P9 P10 AS ATTACK STEADY STATE (DUMMY)
     AND DECAY TIMES
C* * * FOR ENVELOPE N3=108
C* * * TO HAVE P11 AND P12 AS CENTER FREQUENCY AND HALF
BANDWIDTH IN HZ
C* * * * *  FOR FILTER N4=211
C* * * * *  IF THIS IS ALL N=4
C* * * * * * * * * * * * * * * * *
          SUBROUTINE CONVT
          COMMON IP(10),P(100),G(1000)
          IF (G(3).NE.0.0) RETURN
          IF (P(1).NE.1.0) RETURN
          FREQ=511.0/G(4)
          I=P(3)
          NPAR=G(10*I)
          IF (NPAR.EQ.0) GOTO 1
          DO 2 J=1, NPAR
          M=10*I+J
          M=G(M)
          IF (M.GT.200) GOTO 30
          IF (M.GT. 100) GOTO 30
          IF (M.LT.0) GOTO 20
```

```
C* * * * * * * * * * * FREQUENCY
        P(M)=FREQ*P(M)
        GOTO 2
C* * * * * TIME* * * * * ** **
20   M=-M
        P(M)=FREQ/P(M)
        GOTO 2
C* * * * *ENVELOPE* * * * *
30   M=M-100
        P(M+1)=P(4)-P(M)-P(M+2)
        IF (P(M+1))32,33,34
32   P(M)=(P(M)*P(4))/(P(M)+P(M+2))
     P(M+2)=(P(M+2)*P(4))/(P(M))
     P(M+2)=(P(M+2)*P(4))/(P(M)+P(M+2))
33   P(M+1)=128
     GOTO 35
34   P(M+1)=FREQ/(4.0*P(M+1))
35   P(M+2)=FREQ/(4.0*P(M+2))
     P(M)=FREQ/(4.0*P(M))
     GOTO 2
C* * * * *FILTER* * * * * *
40   M=M-200
     D=-(6.2832*P(M+1))/G(4)
     F=(6.2832*P(M))/G(4)
     P(M)=2.*EXP(D)*COS(F)
     P(M+1)=EXP(2.*D)
2    CONTINUE
1    CONTINUE
     RETURN
     END
```


Some GEN subroutines not described in Music V manual have been used in the following examples. Here is given a description of these subroutines, together with the listing.

(A slight change has been performed in Pass III main program to extend the computed GOTO following statement number 3, so that one can provide GEN subroutines GEN6, GEN7, GEN8, GEN9.)

GEN1, GEN2 and GEN3 are as in M. V. Mathews' book, *The Technology of Computer Music*, except for a slight difference in the definition of functions generated by GEN1: in most scores given here, the abscissas range from 1 to 512, while in the book they range from 0 to 511.

GEN4

GEN4 is a Fortran subroutine to generate a stored function as the sum of segments of sinusoids.

The calling sequence is

CALL GEN4

Data is supplied by the P(n), I(n), and IP(n) array:

COMMON I, P/PARM/IP

GEN4 is written in Fortran and requires a sine function SIN (X) which produces the sine of an argument given in radians.

The j^{th} function F_j is generated according to the relation:

$$F_j(i)=(\text{Amplitude Normalizer})\times\sum_{k=1}^{N} A_k \sin\left[\frac{2\Pi}{IP(6)-1}(F_k i+P_k)\right]$$

$i = I_k, I_k + 1, \ldots, J_k$

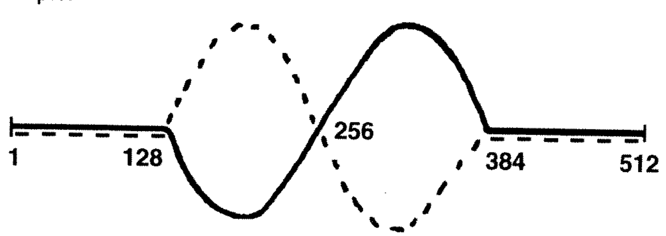
$0 \leq i \leq IP(6) - 1$

The amplitude normalizer is computed so that max $|F_j(i)| = .99999$.

The parameters of the function must be arranged as follows in the data statement:

P(1)	P(2)	P(3)	P(4)	P(5)	P(6)	P(7)	P(8)	P(9)	P(10)	P(11) . . .
GEN	Action Time	4	Function No. (j)	A1	F1	P1	I1	J1	A2	F2

A1 is amplitude
F1 is frequency multiplier
P1 is phase (in samples)
I1 is starting sample
J1 is ending sample
Example:



Continuous line function: GEN, 0, 4, 1, 10*, 2, 0, 128, 384;
(* arbitrary)

Broken line function: GEN, 0, 4, 2, 10*, 2, 256**, 128, 384;

It must be noted that only relative amplitude of components is relevant; the function being normalized, the 10(*) could as well be 1 or 100. On the other hand, phases are expressed in samples (**): 0 corresponds to 0 phase; 128 corresponds to 90° or $\frac{\Pi}{4}$, 256 to 180° or $\frac{\Pi}{2}$ and 384 to 270° or $\frac{3\Pi}{4}$.

In case harmonic partials are used, the first and the last samples of the function are equal: $F_j(0) = F_j[IP(6)-1]$, thus the period in samples is IP(6)-1.

The function is stored starting in I(n) and is scaled by IP(15):

$I(n) = IP(15) * F_j(0)$, etc. where $n = IP(2)+(j-1)*IP(6)$

```
$    FORTRAN LSTOU
CGEN4                        FUNCTION GENERATOR 4
C        * * * * MUSIC V* * * *
C    P(5) AMPLITUDE, P(6) FREQUENCY MULTIPLIER, P(7)
     PHASE, P(8) STARTING SAMPLE, P(9) ENDING SAMPLE
C    SUBROUTINE GEN4
     DIMENSION I(15000), P(100), IP(20), A(7000)
     COMMON I, P/PARM/IP
     EQUIVALENCE (I,A)
     SCLFT=IP(15)
     N1=IP(2)+(IFIX(P(4))-1)*IP(6)
     N2=N1+IP(6)-1
     DO 100 K1=N1,N2
100  A(K1)=0.0
     FAC=6.283185/(FLOAT(IP(6))-1.0)
     NMAX=I(1)-4
```

```
      DO 103 L=5, NMAX, 5
      P2=P(L)
      P3=P(L+1)
      P4=P(L+2)
      JP5=P(L+3)
      IP5=JP5+N1-1
      IP6=IFIX(P(L+4))+N1-1
      DO 105 J=IP5, IP6
      XJ=J-JP5-N1+1
      ARG=XJ*P3+P4
105   A(J)=A(J)+P2*SIN(FAC*ARG)
103   CONTINUE
      XMAX=0.0000001
      DO 115 J=N1,N2
      IF (XMAX-ABS (A(J)))116, 115, 115
116   XMAX=ABS (A(J))
115   CONTINUE
      DO 120 L=N1, N2
120   I(L)=(A(L)*SCLFT*.99999)/XMAX
113   RETURN
      END
```

GEN5

GEN5 is a Fortran subroutine which simply performs various calls in order to skip files or to write an end of file on the output tape.

GEN6

GEN6 is a Fortran subroutine to generate a stored function giving exponential attacks and decays with the ENVelope unit generator.

The calling sequence is

CALL GEN6

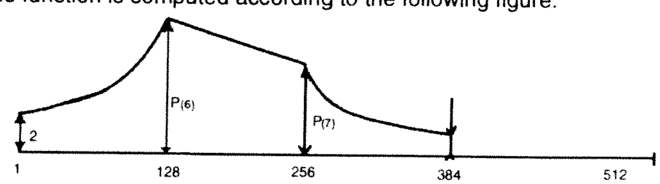
Data is supplied by the P(n), and IP(n) array:

COMMON I, P/PARM/IP

GEN6 is written in Fortran and requires both an exponential and a base 10 logarithmic function: EXP(X), ALOG(X). The parameters of the function are given in the data statement:

P(l)	P(2)	P(3)	P(4)	P(5)	P(6)	P(7)	P(8)
GEN	Action Time	6	Function No. (j)	$-\log_2 A1$	A2	A3	$-\log_2 A4$

The function is computed according to the following figure:



The 1st 128 samples of the function increase exponentially from a value $2^{-P(5)}$ (e.g., 1/2048 if P(5)=11) to a value P(6). They correspond to the attack portion of the envelope generator.

The following 128 samples of the function interpolate linearly between values P(6) and P(7). They correspond to the steady state portion of the envelope generator.

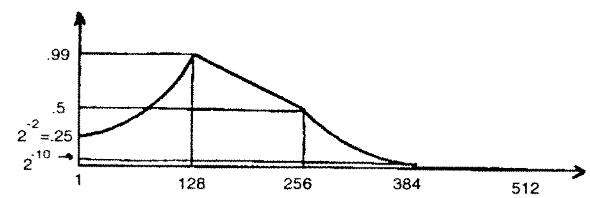
The following 128 samples of the function decrease exponentially from value P(7) to value $2^{-P(8)}$.

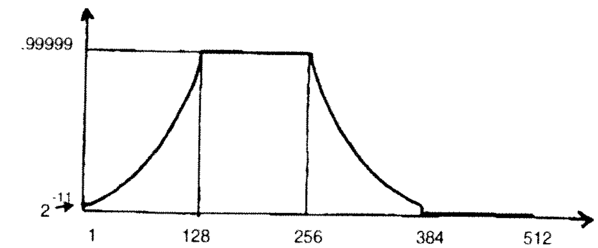
The last 128 samples of the function are zero.

The function is scaled so that its maximum value is .99999. If P(5) or P(8) are zero or negative, the subroutine will give them the default value 2^{-11}.

If P(6) or P(7) are zero or negative, the subroutine will give them the default value .99999.

Examples:






```
CGEN6                    GEN6 FOR ENVELOPE
C    * * * * * MUSIC V* * * * *
CGEN6 FOR ENVELOPE WITH EXPONENTIAL ATTACK AND DECAY
      SUBROUTINE GEN6
      DIMENSION I(15000), P(100), IP(20)
      DIMENSION A(512)
      COMMON I, P/PARM/IP
      SCLFT=IP(15)
      N1=IP(2)+(IFIX(P(4)+0.001)-1)*IP(6)
6     N11=N1-1
      N6=IP(6)
      N2=N6/4
      XNQ=N2-1
      N3=N2+N2
      N4=N3+N2
      ARG1=P(5)
      ARG2=P(6)
      ARG3=P(7)
      ARG4=P(8)
      IF(ARG1) 610, 610, 611
```

```
610  Y1=11.*ALOG(2.)/XNQ
     GOTO 612
611  Y1=ARG1*ALOG(.)/XNQ
612  CONTINUE
     IF(ARG2) 614, 614, 615
614  Y2=.99999
     GOTO 616
615  Y2=ARG2
616  CONTINUE
     IF (ARG3) 618, 618, 619
618  Y3=.99999
     GOTO 620
619  Y3=ARG3
620  CONTINUE
     if (ARG4) 622, 622, 623
622  Y4=11.*ALOG(2.)/XNQ
     GOTO 624
623  Y4=ARG4*ALOG(2.)/XNQ
624  CONTINUE
     DO 630 J=1, N2
     XJ=J-N2
     YJ=Y1*XJ
     A(J)=.99999*EXP(YJ)*Y2
     JJ=J+N11
630  I(JJ)=A(J)*SCLFT
     FACT=(Y3-Y2)/XNQ
     NN2=N2+1
     DO 640 J=NN2, N3
     AJ=J-N2
     A(J)=.99999*(Y2+FACT*AJ)
     JJ=J+N11
```



```
640  I(JJ)=A(J)*SCLFT
     NN3=N3+1
     DO 650 J=NN3, N4
     XJ=NN3-J
     YJ=Y4*XJ
     A(J)=.99999*EXP(YJ)*Y3
     JJ=J+N11
650  I(JJ)=A(J)*SCLFT
     NN4=N1+N4
     NN6=N11+N6
     DO 660 J=NN4, NN6
660  I(JJ)=0
     RETURN
     END
```

<u>GEN7</u>

GEN7 is a Fortran subroutine to generate a stored function which can be a rising exponential, a decaying exponential, or a bell-shaped curve.

The calling sequence is

CALL GEN7

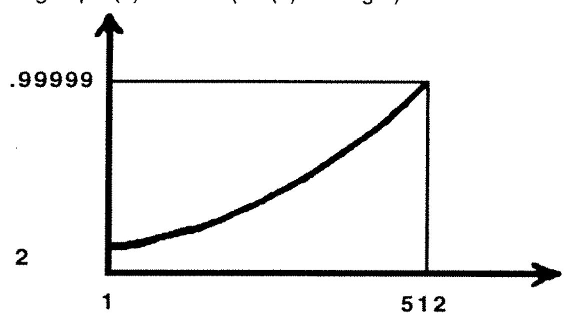
Data is supplied by the P(n), I(n), and IP(n) array:

COMMON I, P/PARM/IP

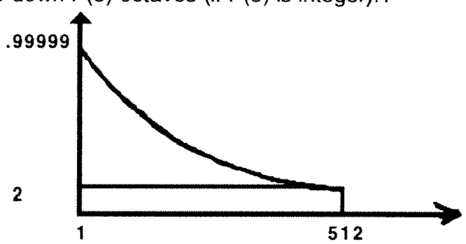
The parameters of the function are given in the data statement:

P(1)	P(2)	P(3)	P(4)	P(5)
GEN Time	Action Number	7	Function	n

If P(5) > 0 GEN7 will compute a function rising exponentially from $2^{-P(5)}$ to .99999. Such a function used to control frequency will cause the pitch to go up P(5) octaves (if P(5) is integer).

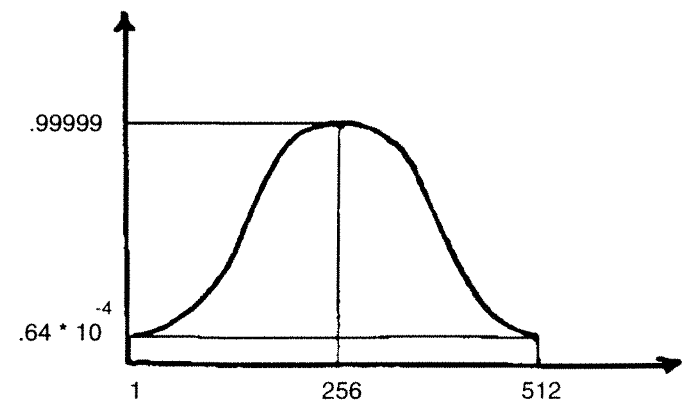


If P(5)<0 GEN7 will compute a function decaying exponentially from .99999 to $2^{-P(5)}$. Such a function used to control frequency will cause the pitch to go down P(5) octaves (if P(5) is integer).1



If P(5)=0 GEN7 will compute a bell-shaped function as represented on the figure. If the ordinate scale is in dB, the curve is a portion of a sine wave with a D.C. bias. The peak ordinate is equal to .99999 and the end points ordinates are equal to $.64 \times 10^{-4}$, which is 84dB below. The formula used is

$$F(x)=\exp\left[\log(.008)\left(1-\cos 2\Pi\left(\frac{x-256.5}{511}\right)\right)\right]$$



CEGGEN7 GEN 7 FOR GLISSANDI OR EXPONENTIAL
DECAYS ************************

```
C
C     IF P(5)=N POSITIVE GO UP N OCTAVES
C     IF P(5)=-N GO DOWN N OCTAVES
C     CONSTANT NUMBER OF SAMPLES PER OCTAVE
C     (EXPONENTIAL PROGRESSION)
C     IF P(5)=0 DRAW SPECIAL SPECTRAL ENVELOPE
      SUBROUTINE GEN7
      DIMENSION I(15000),P(100),IP(20),A(7000)
      COMMON I,P/PARM/IP
      EQUIVALENCE(I,A)
      SCLFT=IP(15)
      N1=IP(2)+(IFIX(P(4))-1)*IP(6)
      N2=N1+IP(6)-1
      DO 100 K=N1,N2
  100 A(K)=0.0
      IF(P(5))200,300,250
C     GO DOWN P(5) OCTAVES
  200 XN= P(5)*ALOG(2.)/511.
      DO 205 J=N1,N2
      XJ=J-N1
      YJ=XN*XJ
      A(J)=EXP(YJ)*.99999
  205 I(J)=A(J)*SCLFT
      GOTO500
C     GO UP P(5) OCTAVES
  250 XN= P(5)*ALOG(2.)/511.
      DO 255 J=N1,N2
      XJ=J-N1-511
      YJ=XN*XJ
```



```
      A(J)=EXP(YJ)*.99999
  255 I(J)=A(J)*SCLFT
      GOTO500
C     AMPLITUDE FOR ENDLESS GLISSANDI
  300 CONTINUE
      DO 325 J=N1,N2
      XJ=J-N1+1
      YJ=(6.2832*(XJ-256.5))/511.
      ZJ=ALOG(.008)*(1.-COS(YJ))
      A(J)=EXP(ZJ)*.99999
  325 I(J)=A(J)*SCLFT
  500 RETURN
      END
```

GEN8

GEN8 is a Fortran subroutine to generate a stored function which can be a bell-shaped curve with one, two or three peaks. The calling sequence is

CALL GEN8

Data is supplied by the P(n), I(n), and IP(n) array:

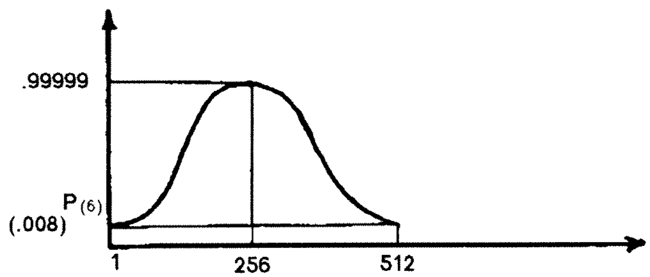
COMMON I,P/PARM/IP

The parameters of the function are given in the data statement:

P(1)	P(2)	P(3)	P(4)	P(5)	P(6)
GEN Time	Action Number	7	Function	n	m

P(6) field is used only if P(5)=0.

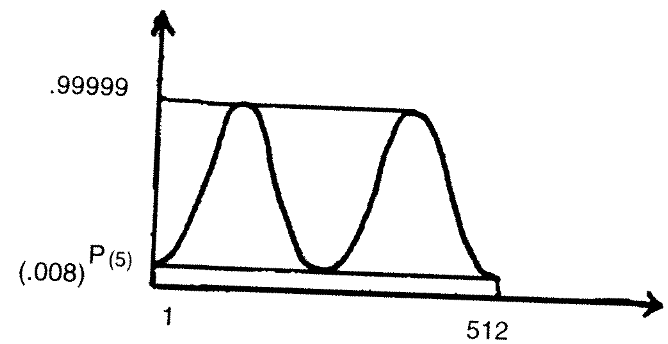
If P(5)=0, GEN8 computes a bell-shaped function with one peak, as shown on the figure. If P(6)=0, default value P(6)=1 is assumed: This corresponds to end points ordinates 42dB below the peaks ordinates.



The function is computed according to the formula

$$F(x) = \exp\left\{\log(.008) \times \frac{P(6)}{2} \times \left[1-\cos\left(2\Pi\frac{x-256.5}{511}\right)\right]\right\}$$

If P(5)>0, GEN8 computes a bell-shaped function with two peaks, as shown on the figure.

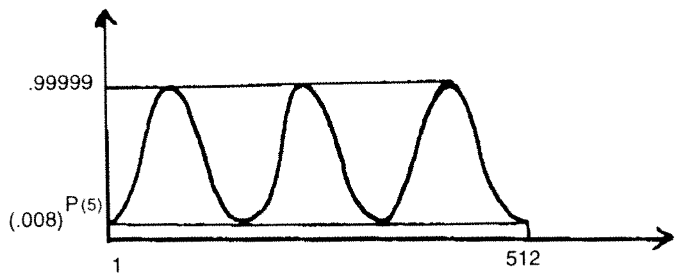


The function is computed according to the formula:

$$F(x) = \exp\left\{\log(.008) \times \frac{P(5)}{2} \times \left[1-\sin 2\Pi\left(\frac{x-65}{256}\right)\right]\right\}$$

For P(5)=1, the end points ordinates are 42 dB below the peaks ordinates.

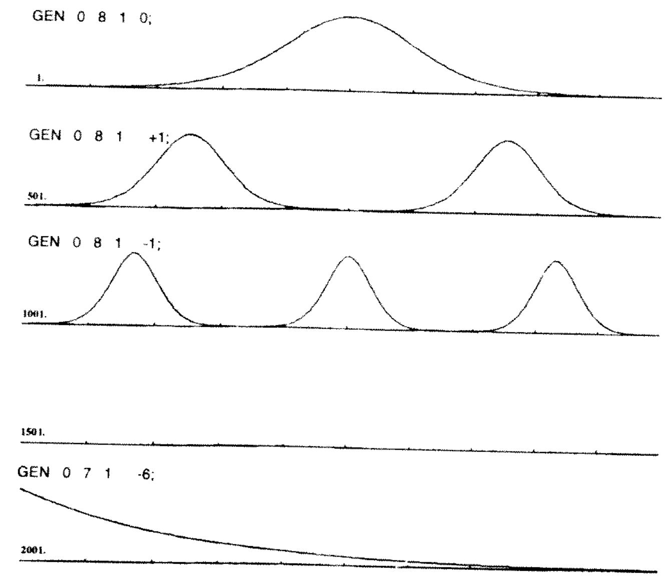
If P(5)<0, GEN8 computes a bell-shaped function with three peaks, as shown on the figure.



The function is computed according to the formula:

$$F(x) = \exp\left\{\log(.008) \times \frac{-P(5)}{2} \times \left[1 - \sin 2\Pi\left(\frac{x-44}{170}\right)\right]\right\}$$

For P(5)=-1, the end points ordinates are 42dB below the peaks ordinates.



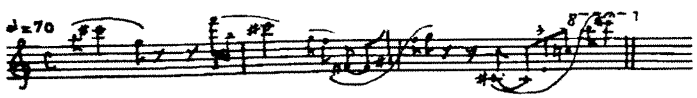
```
CENVGEN8                    GEN8 FOR 1,2,3 PEAK CURVES
                                     ***********
C
C     FOR TONE HEIGHT VERSUS TONALITY STUDY
C     P(5) ZERO    ONE PEAK
C     P(5) POSITIVE   TWO PEAKS
C     P(5) NEGATIVE   THREE PEAKS
      SUBROUTINE GEN8
      DIMENSION I(15000),P(100),IP(20),A(7000)
      COMMON I,P/PARM/IP
      EQUIVALENCE(I,A)
      SCLFT=IP(15)
      N1=IP(2)+(IFIX(P(4))-1)*IP(6)
      N2=N1+IP(6)-1
      DO 100 K=N1,N2
  100 A(K)=0.0
      IF(P(5))200,250,300
C     THREE PEAKS
  200 CONTINUE
      XM=-P(5)
      DO 225 J=N1,N2
      XJ=J-N1+1
      YJ  =(6.2832*(XJ-44.))/170.
      ZJ  = ALOG(.008)*(1.-SIN(YJ))*.5*XM
      A(J)=EXP(ZJ)
  225 I(J)=A(J)*SCLFT
      GOTO500
C     ONE PEAK
  250 CONTINUE
      XM=P(6)
      IF(XM.EQ.0.)XM=1.
      DO 275 J=N1,N2
```



```
      XJ=J-N1+1
      YJ  =(6.2832*(XJ-256.5.))/511.
      ZJ  = ALOG(.008)*(1.-COS(YJ))*.5*XM
      A(J)=EXP(ZJ)
  275 I(J)=A(J)*SCLFT
      GOTO500
C     TWO PEAKS
  300 CONTINUE
      XM=P(5)
      DO 325 J=N1,N2
      XJ=J-N1+1
      YJ  =(6.2832*(XJ-65.))/256.
      ZJ  = ALOG(.008)*(1.-SIN(YJ))*.5*XM
      A(J)=EXP(ZJ)
  325 I(J)=A(J)*SCLFT
  500 RETURN
      END
```

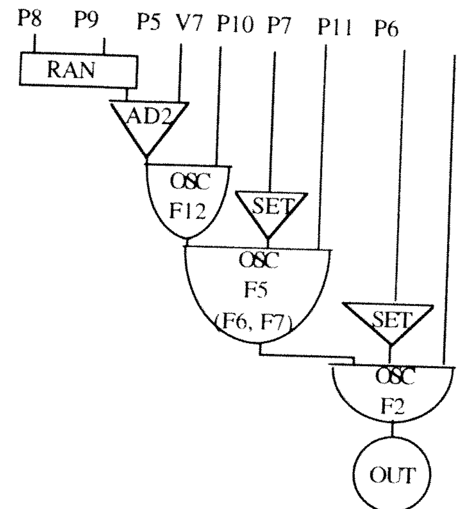
<u>#100 (Track 18)</u>

This is a melody played by an instrument reminiscent of a flute.



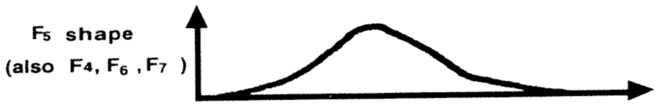
<u>Instrument#2</u>

(diagram next page)



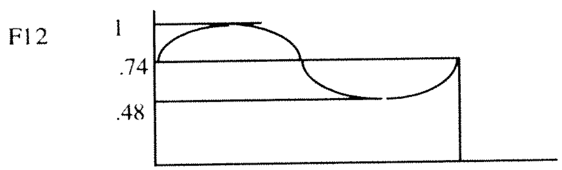
This instrument gives a wave with a time envelope, a random amplitude modulation and a periodic amplitude modulation.

The wave corresponds to function F2; by means of SET, it can be changed to function Fn, where n is the value of P11 in the note card. (If n < 0 no change is effected.) Here F1 - sine wave, F2 - 4 harmonics wave, F3 - 6 harmonics wave - are used; for these 3 functions the fundamental is dominant. The function with the richest harmonic content is used for the lowest note. Care is taken to avoid foldover, except at action time 9 and 9.1 where a small amount of foldover is deliberately introduced. The time envelope corresponds to functions F5 and also F4, F6, F7; it gives slow attack and decay.



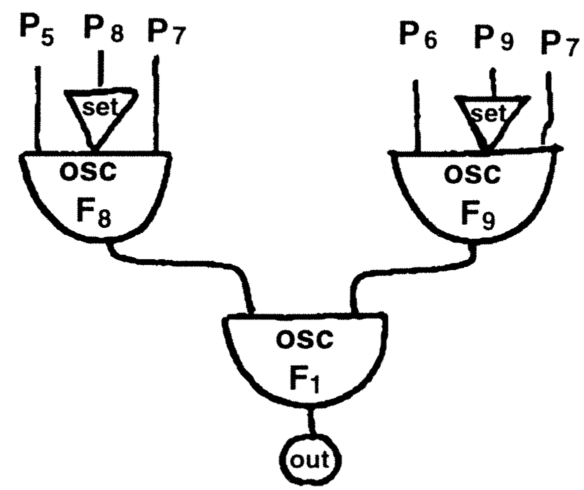
The random amplitude modulation range is only 1% of the amplitude, (see CONVT), and its rate is around 60Hz. This modulation is only marginally significant.

The periodic amplitude modulation is performed using function F12, giving a sine wave with a D.C. bias. The rate is given by V7 and is around 5Hz. Both F12 and V7 are changed in the course to the melody, (similarly to other parameters) to give more naturalness to the melody.



<u>Instrument #3</u>

This instrument is simply used to introduce frequency glides controlled by F9 or to increase the proportion of fundamental (in conjunction with instrument 2).



Note: Since only this example was to be generated with this type of tone quality, I have not attempted to improve the code in terms of economy of specification.

```
COMMENT:--------------------JCR100-----------;
COMMENT FLUTE RUN ON TAPE M1669 FILE 6; GEN 0 5 5;
COMMENT: SAMPLING RATE 10000 HZ; SIA 0 4 10000;
COMMENT: INSTRUMENTS FOR FLUTE LIKE TONES;
INS 0 2;RAN P8 P9 B3 P30 P29 P28;AD2 B3 P5 B3;
OSC B3 V7 B3 F12 P25;SET P10;
OSC B3 P7 B3 F5 P27;SET P11;IOS B3 P6 B3 F2 P26;OUT B3
B1;END;
INS 0 3;SET P8;OSC P5 P7 B3 F8 P30;SET P9;OSC P6 P7 B4 F9 P29;
OSC B3 B4 B3 F1 P28;OUT B3 B1;END;
COMMENT: METRONOME MARKING 70;
SV2 0 2 30; SV2 0 30 0 70 20 70;
SV3 0 7 .24;
GEN 0 2 1 1 1;
GEN 0 2 2 1 .2 .08 .07  4;
GEN 0 2 3 1 .4 .2 .1 .1 .05 6;
GEN 0 1 4 0 1 .2 50 .6 140 .99 180 .9 205 .5 250 .25 300 .12 350
 .06 400 .03 450 0 512;
GEN 0 1 5 0 1 .2 50 .6 150 .99 200 .2 350 0 512;
GEN 0 1 6 0 1 .2 50 .5 250 .2 350 0 512;
GEN 0 1 7 0 1 .5 80 .5 140 .99 160 .4 280 .6 420 0 512;
GEN 0 1 8 0 1 .4 150 .99 350 .5 400 .24 450 0 512;
GEN 0 1 9 .895 1 .99 512;
GEN 0 1 10 .999 1 .999 512;
GEN 0 2 12 .26 .74 1;
NOT .88 3 .12 1200 988 .12 8 10;
NOT 1 2 2 800 1109 2 20 60;
GEN 1 1 8 0 1 .99 100 0 512;
NOT 1 3 .7 300 1107 .7 8 10;
GEN 3 2 12 .3 .6 1;
NOT 3 2 .8 300 784 .5 30 50 4;
```



```
NOT 4.5 3 .375 1200 1397 .375 5;
NOT 4.85 3 .15 1200 992 .15;
NOT 5 3 .7 300 1100 .7;
NOT 5.01 2 2 1200 1109 2 30 80 6 2;
NOT 7 2 .2 400 784 .2 40 70 7;
NOT 7.2 2 .3 300 698 .3 30 60 5;
NOT 7.51 2 1 300 370 1 30 50 6 2;
NOT 7.5 3 .5 150 368 .5 8;
NOT 8.5 2 .5 400 415 .5 50 60 5;
NOT 9 2 .12 900 1396 .6 30 80 4 2;
NOT 9.1 2 1.2 900 1558 .8 30 90 4 2;
SV3 10.24 7 .31;
NOT 10.25 2 1.08 900 277 1.08 40 60 7 3;
SV3 12.08 7 .28;
NOT 10.25 3 1 200 275 1 6 10;
NOT 11.35 2 .36 500 329 .31 30 60 5 2;
NOT 11.72 2 .36 800 528 .26 30 60 5 2;
NOT 12.09 3 .20 950 2217 .2 6 9;
NOT 12.10 2 .15 700 1975 .13 40 90 5 1;
NOT 12.23 2 2.5 999 2217 1 40 90 4 1;
TER 19;
```

```
C           CONVT FOR FLUTE
      SUBROUTINE CONVT
      COMMON IP(10),P(100),G(1000)
      IF(P(1).NE.1.)GOTO100
      F=511./G(4)
      P(6)=F*P(6)
      P(7)=F/P(4)
```

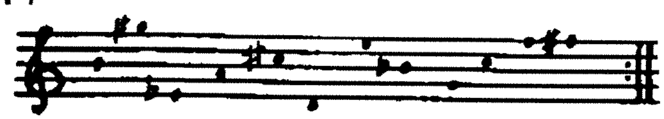
```
      IF(P(3).EQ.3.)GOTO100
      P(9)=F*P(9)
      P(8)=.01*P(5)
100   RETURN
      END
```


#150 (Track 19)

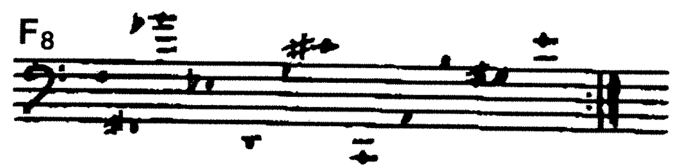
This run gives a serial excerpt. It makes use of three different tone qualities, particularly one obtained through non-linearity which has some similarity with clarinet sounds. The 12-tone development is done automatically in the 3rd pass, by repeated scanning of stored functions: frequency controlling functions, corresponding to the pitch rows, and amplitude controlling functions, corresponding to rhythm and accent rows; this way hundreds of musical notes are generated from only 10 note cards, using a process similar to M. V. Mathews' cyclic algorithm but performed in the 3rd pass[6].

The example uses two pitch rows, specified respectively by F7 and F8. The frequency input of the oscillators using these functions is the frequency of the highest note in the row. The rows are as follows:

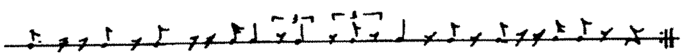
F7



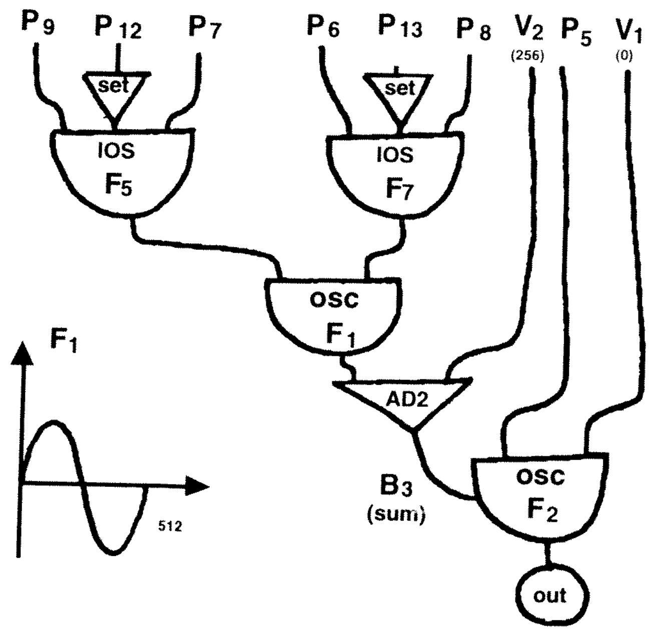
F8



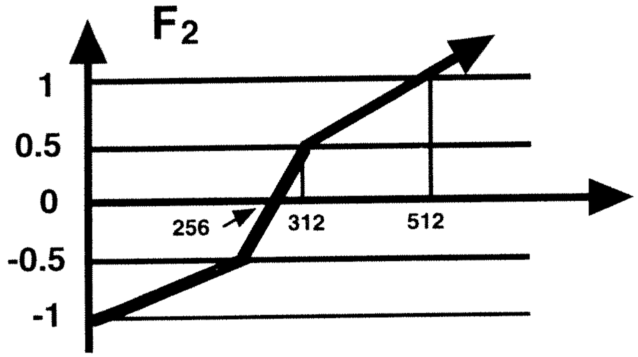
The example uses two rhythm rows, specified by F5 and F6. For instance F5 corresponds to the following rhythm:



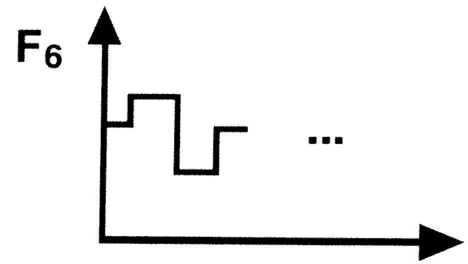
The tempo is ♩ =132 for a scanning duration of 5.91s.
The oscillators IOS using F5, F6, F7, F8 accept negative increments that will cause the function to be scanned backwards.
(Diagram next page.)



Instrument #1



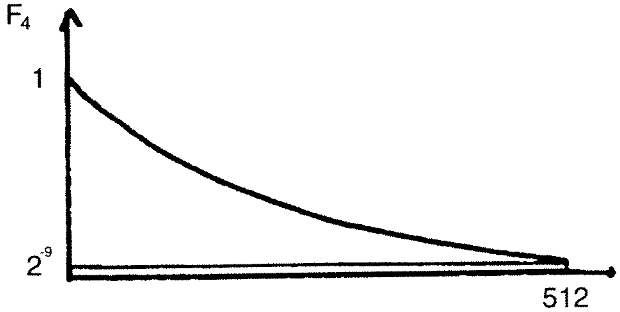
Instrument #1



Instrument #1 is diagrammed here. A sine wave F1 is amplitude and frequency controlled by functions F5 and F6 respectively, and then submitted to a non-linear transfer, accordmg to the characteristics of function F2. F2 x P5 gives the output as a function of the input B3, which must be in the interval -256,+256. This is achieved by using the bottom oscillator in a degenerate way, whereby B3 is used as sum with a frequency increment of 0(V1=0). Both P9 and P5 determine the maximum amplitude, but the value of P9 (in the interval -256,+256) determines the amount of "distortion" performed on the sine wave. (The output is still a sine wave if P9<312-256 = 56). The distortion generates odd harmonics so that a sampling rate of 20,000 Hz had to be used to avoid objectionable foldover for fundamental frequencies around 1500 Hz.

Instrument #2 simply generates a waveform defined by F3, amplitude and frequency controlled respectively by functions F6 and F8. F3 is an all-positive waveform, generated by GEN7 (cf. description): it is a sine wave in a dB scale, with the lowest point 84 dB below the highest point. There is a marked difference between the aural effect of this wave and that of a true sine wave.

Instrument #3 generates a sine wave whose frequency is controlled by F7 and whose amplitude is controlled by F4. F4 is a decaying exponential: this gives a percussive sound. The rate of scanning the amplitude function is about 12 times the rate of scanning the frequency function: if it were exactly that, it would give one "stroke" per pitch. By divorcing the rates of scanning for amplitude and for frequency functions, one can obtain repeated pitches or legato transitions between pitches.



Note: In the printout for run #150, semicolons (;) are replaced by dollar signs ($). (This run was performed on a machine without a (;) in the character set). Also GEN1 is for this example defined with abscissas ranging from 0 to 511.

```
COMMENT:--------------------JCR100-----------;
COMMENT  SERIAL FUNCTIONS WITH NON LINEARITY$
COMMENT FCTS SERIELLES CLAR 1 CELLO 2$
INS 0 1$SET P12$IOS P9 P7 B3 F5 P10$
```

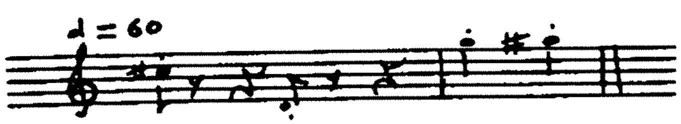


```
SET P13$IOS P6 P8 B4 F7 P11$
OSC B3 B4 B3 F1 P30$AD2 B3 V2 B3$
OSC P5 V1 B2 F2 B3$OUT B2 B1$END$
GEN 0 2 1 1 1$
GEN 0 1 2 -.99 0 -.5 200 .5 312 .99 512$
INS 0 2$SET P12$IOS P5 P7 B3 F6 P10$
SET P13$IOS P6 P8 B4 F8 P11$
OSC B3 B4 B3 F3 P30$OUT B3 B1$END$
SV3 0 1 0 256$
GEN 0 7 3 0 1$
GEN 0 1 4 0 0 .2 80 140 .99 250 .8 330 .2 400 0 511$
COMMENT FCTS DE RYTHME PUIS DE PITCH$
GEN 0 1 5 0 0 .99 4 .6 10 0 29 0 58 .7 62 .3 72 0 78 0 97 .4 102
 .1 110 0 118 0 147 .9 149 .6 153 .2 156 .8 158 .3 180 0 196 0 209
 .3 211 .7 225 .3 230 0 233 0 248 .4 250 .9 255 0 262 0 275 .99 276
 0 314 0 334 .8 335 .2 345 0 353 0 373 .99 375 0 393 0 422 .6 423 .1
432 .7 433 .7 440 0 452 0 511$
GEN 0 1 6 0 0 .9 3 .6 50 .2 56 0 58 0 84 .8 86 .9 107 0 114 0 151 .99
155 .3 167 0 171 0 211 .99 213 .3 224 .1 227 .99 231 .6 150 .8 280 .05
284 .5 286 .4 290 .05 292 .6 300 .8 306 0 312 0 381 .6 383 0 397 0 410
.7 415 .2 425 0 440 0 492 .99 496 .6 504 0 511$
GEN 0 1 7 .593 0 .593 42 .99 43 .99 85 .37 86 .37 127 .527 128 .527
170 .667 171  .667 212  .351 213 .351 255 .790 256 .790 298 .555 299
.555 340 .468 341 .468 383 .624 384 .624 425 .832 426 .832 468 .889
469 .889 511 $
GEN 0 1 8 .375 0 .375 42 .200 43 .200 85 .999 86 .999 127 .333 128
.333 170 .158 171 .158 212 .422 213 .422 255 .601 256 .601 298 .141
299 .141 340 .237 341 .237 383 .534 384 .534 425 .450 426 .450 468
.712 469 .712 511$
INS 0 3$OSC P5 P7 B3 F4 P30$IOS P6 P8 B4 F7 P29$
OSC B3 B4 B3 F1 P28$OUT B3 B1$END$
```

```
GEN 0 7 4 -9$
SIA 0 4 20000$
COMMENT TO SET GENERAL CONVT$
SV2 0 10 3 6 -7 -8$
SV2 0 20 3 6 -7 -8$
SV2 0 30 3 6 -7 -8$
NOT 1 1 11.82 250 1640 11.82 10.90 250$
NOT 24.62 1 23.64 200 1640 -11.82 -10.90 250$
NOT 36.44 1 11.82 250 2304 11.82 10.90 250$
NOT 3.72 2 16.32 800 228 8.16 10.90$
NOT 6.45 1 29.6 450 820 5.91 -5.45 220$
NOT 6.45 1 29.6 400 576 2.955 -5.45 200$
NOT 24 2 16.32 300 455 -16.32 -10.90$
NOT 24 1 23.64 400 820 -11.82 10.90 250$
NOT 30 2 16.32 800 228 8.16 10.90$
NOT 30 3 22 500 6560 .92 11$
TER 54$
```

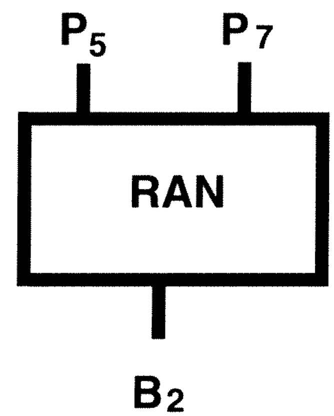
#200 (Track 20)

This run provides a few "brass-like" tones. Here no attempt has been made towards economy of specification: schematized data from real trumpet tones have been used (cf., J. C. Risset and M. V. Mathews, Physics Today, Feb.1969).



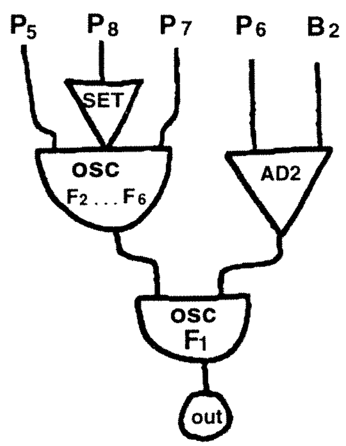
Instrument #1

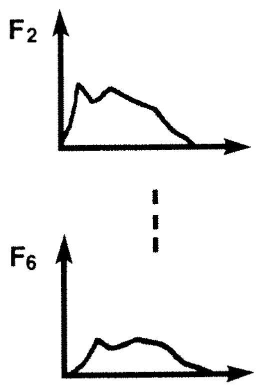
This is a degenerate instrument which simply provides for random frequency modulation in the other instruments. The range is around .5% of the fundamental frequency; the rate is around 10Hz. (These are low values: in this example the random modulation is not very significant.)



Instruments #2, 3, . . ., 6

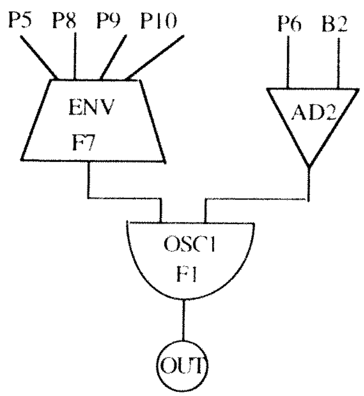
These instruments are used to synthesize tones with different envelopes for each partial: 1st partial with function F2, . . ., 5th partial with function F6. Random frequency modulation is possible using instrument #1. F1 is a sine wave.





Instrument #7

This instrument is used to synthesize partials with different attacks and decay.




```
COMMENT -------------------JCR200----------------------;
COMMENT:BRASSY TONES WITH INDEPENDENT CONTROL OF
THE HARMONICS;
COMMENT: TAPE M2029;
COMMENT: SAMPLING RATE 12.5 KC;
COMMENT:FOR FREQUENCY MODULATION AT RANDOM;
INS 0 1;RAN P5 P7 B2 P30 P29 P28;END;
```

```
COM FOR INDEPENDENT HARMONIC CONTROL;
INS 0 2;SET P8;OSC P5 P7 B3 F2 P30;
AD2 P6 B2 B4;OSC B3 B4 B3 F1 P29;OUT B3 B1;END;
INS 0 3;SET P8;OSC P5 P7 B3 F3 P30;
AD2 P6 B2 B4;OSC B3 B4 B3 F1 P29;OUT B3 B1;END;
INS 0 4;SET P8;OSC P5 P7 B3 F4 P30;
AD2 P6 B2 B4;OSC B3 B4 B3 F1 P29;OUT B3 B1;END;
INS 0 5;SET P8;OSC P5 P7 B3 F5 P30;
AD2 P6 B2 B4;OSC B3 B4 B3 F1 P29;OUT B3 B1;END;
INS 0 6;SET P8;OSC P5 P7 B3 F6 P30;
AD2 P6 B2 B4;OSC B3 B4 B3 F1 P29;OUT B3 B1;END;
COMMENT:FOR ATTACK AND DECAY TIMES;
INS 0 7;ENV P5 F7 B3 P8 P9 P10 P30;AD2 P6 B2 B4;
OSC B3 B4 B3 F1 P29;OUT B3 B1;END;
SIA 0 4 12500;
GEN 0 2 1 1 1;
GEN 0 1 2  0 1 .001 10 .282 26 .112 40 .178 429 .159 473 .008 500 .001
512;
gen 0 1 3 0 1 .001 19 .500 434 .355 454 .016 490 .001 512;
GEN 0 1 4 0 1 .001 23 .55 435 .001 512;
GEN 0 1 5 0 1 .001 10 .005 19 .224 418 .224 431 .178 458 .001 512;
GEN 0 1 6 0 1 .001 10 .009 21 .089 33 .022 45 .022 73 .112 226 .178
264 .071 345 .062 468 .001 512;
GEN 0 1 7 0 1 .99 128 .999 256 0 384 0 512;
NOT 1 1 .17 .6 554 10;
NOT 1 7 .17 200 554 0 7.5 0 140;
NOT 1 7 .17 160 1108 0 7.5 0 110;
NOT 1 7 .17 350 1662 0 12 0 85;
NOT 1 7 .15 310 2216 0 14 0 80;
NOT 1 7 .14 160 2770 0 24 0 65;
NOT 1 7 .14 200 3324 0 27 0 60;
```



```
NOT 1 7 .14 99 3878 0 32 0 60;
NOT 1 7 .14 200 4432 0 30 0 60;
NOT 1 7 .14 80 4986 0 35 0 60;
NOT 3 1 .15 .6 293 10;
NOT 3 8 .15 50 293 0 10 0 140;
NOT 3 7 .15 80 586 0 10 0 110;
NOT 3 7 .15 100 879 0 12 0 85;
NOT 3 7 .15 175 1172 0 17 0 80;
NOT 3 7 .15 180 1465 0 25 0 65;
NOT 3.01 7 .15 150 1758 0 30 0 60;
NOT 3.01 7 .15 100 2051 0 35 0 60;
NOT 3.01 7 .13 80 2344 0 35 0 60;
NOT 3.01 8 .14 50 2637 0 40 0 100;
NOT 3.01 8 .14 80 2930 0 40 0 100;
NOT 3.01 8 .14 140 3223 0 45 0 100;
NOT 3.01 8 .13 90 3516 0 45 0 100;
NOT 3.01 8 .13 45 3809 0 40 0 100;
NOT 3.01 8 .13 25 4102 0 45 0 90;
NOT 5 1 .4 .4 784 20;
NOT 5 2 .4 800 784;
NOT 5 3 .4 800 1568;
NOT 5 4 .4 800 2352;
NOT 5 5 .4 800 3136;
NOT 5 6 .4 800 3920;
NOT 6 1 .7 .4 830 20;
NOT 6 2 .7 1400 830;
NOT 6 3 .7 1000 1660;
NOT 6 4 .7 1000 2490;
NOT 6 5 .7 1000 3320;
NOT 6 6 .7 1000 4150;
TER 8;
```

```
      SUBROUTINE CONVT
      COMMON IP(10),P(100),G(1000)
      IF(P(1).NE.1.)GOTO100
      F=511./G(4)
      P(6)=F*P(6)
      IF(G(10).GE..5)GOTO200
      IF(P(3).EQ.1.)GOTO10
      IF(P(3).EQ.7.)GOTO70
      P(7)=F/P(4)
      GOTO100
10     P(5)=.01*P(5)*P(6)
      P(7)=F*P(7)
      GOTO100
70     FENV=F*.25
      P(8)=.001*P(8)
      P(9)=.001*P(9)
      P(10)=.001*P(10)
      P(9)=P(4)-P(8)-P(10)
      IF(P(9))2,3,4
2      P(8)=(P(8)*P(4))/(P(8)+P(10))
      P(10)=(P(10)*P(4))/(P(8)+P(10))
3     P(9)=128.
      GOTO5
4     P(9)=FENV/P(9)
      P(8)=FENV/P(8)
      P(10)=FENV/P(10)
      GOTO100
200          P(6)=P(6)*P(3)*10.
      P(7)=F/P(4)
      FENV=F*.25
      P(8)=.001*P(8)
```



```
      P(9)=.001*P(9)
      P(10)=.001*P(10)
      P(9)=P(4)-P(8)-P(10)
      IF(P(9))202,203,204
202          P(8)=(P(8)*P(4))/(P(8)+P(10))
      P(10)=(P(10)*P(4))/(P(8)+P(10))
203          P(9)=128.
      GOTO205
204          P(9)=FENV/P(9)
205          P(8)=FENV/P(8)
      P(10)=FENV/P(10)
 100  RETURN
      END
```

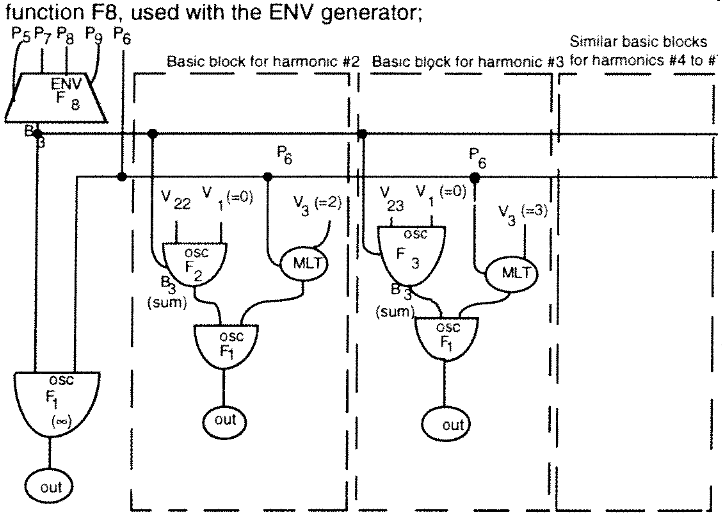
<u>#201 (Track 21)</u>

This is the same run as #200, but played back at a sampling rate of 5000Hz instead of 12,500Hz--hence all frequencies are multiplied by .4, all durations are multiplied by 2.5.

<u>#210 (Track 22)</u>

This run gives some examples of brass-like sounds synthesized with more economy of specification than in #200, using an instrument designed to produce sounds whose spectra depend upon the amplitude of one component (cf. J. C. Risset and M. V. Mathews, <u>Physics Today</u>, Feb. 1969). It must be noted that this instrument is by no means limited to the production of this type of sounds: the components need not be harmonically related and the functions used can be entirely different.

The instrument (#1) is diagrammed here. The amplitude of one partial (which will be in this example harmonic #1) is controlled by function F8, used with the ENV generator;



the maximum amplitude of this component is determined by P5, and has to be smaller than 512. The output of the ENV generator is input-output block B3. B3 is used as amplitude input of the oscillator generating the contribution of harmonic #1, and also for another purpose. Hence B3 has to be reserved in this instrument, it cannot for example be used as output for another oscillator. For each harmonic, B3 is used as sum for an oscillator with a frequency increment of 0 (V1=0), which performs

simply as a function look-up unit. E.g., for harmonic #2 the output of this oscillator will be the product of V22 by the value of stored function F2 for an abscissa equal to the current value stored in B3 (e.g., the current value of the output of the ENV generator); this output is stored into B4 and used as amplitude input for the oscillator generating the contribution of harmonic #2--the frequency input being the product of the fundamental frequency P6 by the constant V2 = 2. Hence the value of the amplitude of harmonic #2 is a prescribed function of the amplitude of harmonic #1 (this function being determined by V22 and by F2). Similar basic blocks of unit generators give, in a similar way, the amplitude of each harmonic as a prescribed function of the amplitude of harmonic #1. Hence the spectrum of the sound depends upon the amplitude of harmonic #1.

Fundamental frequency is given in Hz by P6. Attack time and decay time are given in s by P7 and P9; the CONVT subroutine computes the steady state time as P4-P7-P9.

The example includes two sections, which differ by the constants V22,. . ., V28 and the functions F2, . . ., F7 used with the instrument. Hence the way the spectrum depends upon the amplitude of harmonic #1 is different in the two sections; however in both sections it retains one important characteristic of brassy tones, namely the fact that the proportion of high frequency energy increases with the intensity of the sound.

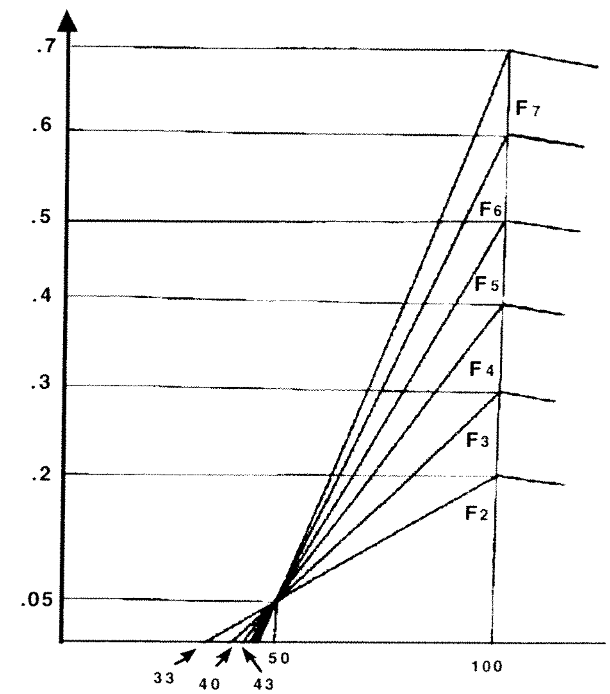
In the first section functions F2 to F7 are as plotted (Plot attached). All functions have value .05 for abscissa 50: when 1st harmonic's amplitude is 50, 2nd harmonic's amplitude is V22 x .05, (in this case 1000 x .05 = 50), 3rd harmonic's amplitude is V23 x .05, and so on. Thus when P5 = 50, the amplitude of the successive harmonics are proportional to 1000, V22, V23, . . . When P5 increases from value 50, due to the functions used the contributions of harmonics #2, 3, . . ., 7 increase respectively 2, 3, . . ., 7 times as fast. Overloading (peak amplitude higher than 2048) occurs when P5 is between 80 and 90. So the useful range for P5 is from 0 to 80 (but the sound is sinusoidal for P5 <33).

The first sound is a long tone with dynamics represented by sf-p-cresc, to illustrate how the spectrum brightens when the amplitude increases. Then follow 9 short sounds of varied amplitude, with a large amplitude overshoot at the beginning of the sound. The attack time used is 50 ms (larger than in most actual trumpet sounds--because of the unusual way the harmonics come in).

In the second section, slightly different functions F2 and F3 and different values for V22 to V27 are used. The section comprises five sustained notes and one crescendo note.

It is useful to add to the instrument an MLT generator to scale the output by a factor specified on the note card: this permits the instrument to be played by several voices, with P5 = 80 for each of them, without overloading (merely reducing the value of P5 for each voice would change the spectrum).

The sounds of this example are not presented as good imitations of trumpet sounds: the spectrum is not reproduced accurately and in particular there is no formant structure; there are not enough components, and there is no frequency control (both formant structure and frequency control could of course be incorporated in this type of instrument). On the other hand, using this type of control of the spectrum, one can obtain somewhat "brassy" sounds with only 3 functions, one controlling harmonics 2 and 3, the second one controlling 4 and 5, and the third one controlling 6 and 7. Also, as mentioned before, the utility of this type of spectrum variation is not limited to brass-like sounds.



Plot of Functions for #210

Note: In the printout for run #210, semicolons (;) are replaced by dollar signs ($) (this run was performed on a machine without a ; in the character set). Also GEN1 is for this example defined with abscissas ranging from 0 to 511.

```
COMMENT -------------------JCR210-------------------;
COMMENT: SIMPLIFIED BRASSY SOUNDS$
INS 0 1$ENV P5 F8 B3 P7 P8 P9 P30$OSC B3 P6 B4 F1 P29$OUT B4
B1$
OSC V22 V1 B4 F2 B3$MLT P6 V2 B5$OSC B4 B5 B4 F1 P27$OUT B4
B1$
OSC V23 V1 B4 F3 B3$MLT P6 V3 B5$OSC B4 B5 B4 F1 P25$OUT B4
B1$
OSC V24 V1 B4 F4 B3$MLT P6 V4 B5$OSC B4 B5 B4 F1 P23$OUT B4
B1$
OSC V25 V1 B4 F5 B3$MLT P6 V5 B5$OSC B4 B5 B4 F1 P21$OUT B4
B1$
OSC V26 V1 B4 F6 B3$MLT P6 V6 B5$OSC B4 B5 B4 F1 P19$OUT B4
B1$
OSC V27 V1 B4 F7 B3$MLT P6 V7 B5$OSC B4 B5 B4 F1 P17$OUT B4
B1$END$
COMMENT TO SET GENERAL CONVT$SV2 0 10 2 6 107$
SV3 0 2 2 3 4 5 6 7$
SV3 0 22 1000 1000 2000 1900 1250 1000 850$
GEN 0 2 1 1 1$
GEN 0 1 2 0 0 0 33 .2 100 0 511$GEN 0 1 3 0 0 0 40 .3 100 0 511$
GEN 0 1 4 0 0 0 43 .4 100 0 511$GEN 0 1 5 0 0 0 45 .5 100 0 511$
GEN 0 1 6 0 0 0 46 .7 100 0 511$GEN 0 1 7 0 0 0 46 .7 100 0 511$
GEN 0 1 8 0 0 .5 128 1.999 256 0 384 0 511$
NOT 1 1 5 80 554 .05 5 .25$
GEN 7 1 8 512 0 0 .99 100 .65 110 .8 128 .6 256 .3 300 0 383 0 511$
NOT 8.5 1 .3 45 554 .05 0 .1$
```

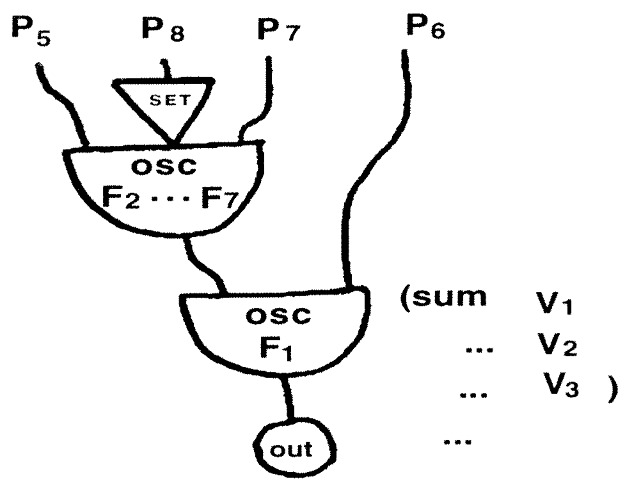


```
NOT 9 1 .3 50 554 .05 0 .1$
NOT 9.5 1 .3 55 554 .05 0 .1$
NOT 10 1 .3 80 554 .05 0 .1$
NOT 10.5 1 .3 60 554 .05 0 .1$
NOT 11 1 .2 65 554 .05 0 .1$
NOT 11.5 1 .2 70 554 .05 0 .1$
NOT 12 1 .2 75 554 .05 0 .1$
NOT 12.5 1 .3 80 554 .05 0 .1$
SEC 14$
SV3 0 22 1000 4000 5000 2400 4000 1000 5000$
GEN 0 1 2 0 0 0 33 .05 50 .11 100 0 511$
GEN 0 1 3 0 0 0 40 .05 50 .12 100 0 511$
GEN 0 3 8 0 45 85 0 0$
NOT 1 1 .6 50 682 .05 .4 .15$
NOT 2 1 .6 60 682 .05 .4 .15$
NOT 3 1 .6 70 682 .05 .4 .15$
NOT 4 1 .6 80 682 .05 .4 .15$
NOT 5 1 3 85 682 .05 3 .2$
TER 9$
```

#250 (Track 23)

This run gives an example of how the same waveshape can give different tone qualities, depending upon the amplitude envelope; here are presented sounds which could be described as "reedy" (like oboe or bombarde sounds) or "plucked" (like harpsichord sounds). Also an example of "choral effect" is given.

Instruments #1, 2, 3



These instruments are diagrammed here. They give waveshape F1 with an envelope defined by functions F2 to F7. The sum of the wave-

shape oscillator is stored in a Pass III variable: this permits click-free "legato" transitions between successive notes (that is, transitions where the amplitude does not go to 0 at the end or the beginning of the note).

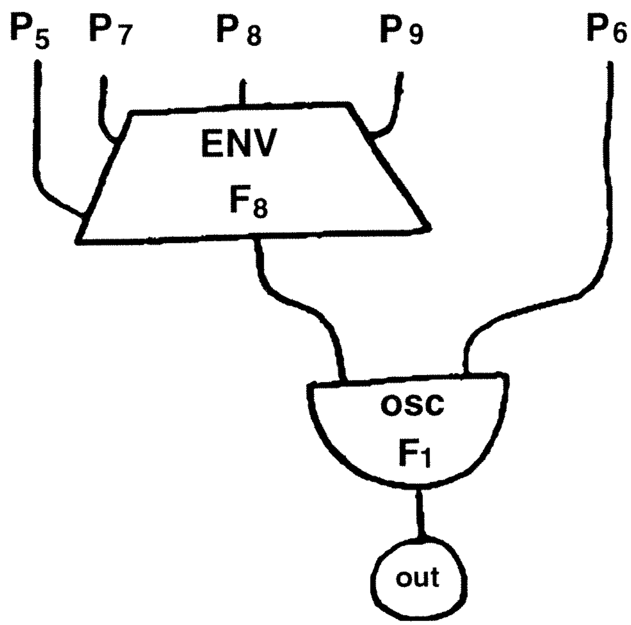
Instruments #1, 2, and 3 are defined by functions F2, F3 and F2 respectively for the envelope: but these function numbers can be modified in the note card, using SET. The functions used insure long attacks and decays (longer than 50 milliseconds) and "legato" transitions between successive notes.

Function Fl comprises 11 harmonics. This same function is used in instrument #4.

The first section plays an excerpt of a Brittany folk melody with one voice (produced by instrument #1). The scale is not equally tempered; the leading tone is conspicuously low. The tone quality is reminiscent of a double reed instrument.

The second section plays a similar melody, but with three voices played by instruments 1 to 3. The frequency and time differences between the voices (up to a several per cent difference in frequency and up to .08s in time) somewhat evocates the sound of a number of players (choral effect). (From a single voice, the additional voices note cards could be generated automatically by use of a simple PLF subroutine.)

The third section plays a related melody with the same waveshape F1 but with a short (exponential) attack and an exponential decay. This section uses instrument #4, diagrammed here. Function F8 gives an exponential attack and decay between 1 and 2^{-9}. The notes of this section have no steady state, an exponential attack time of 10 milliseconds which corresponds to a very sharp attack, and an exponential decay time varying between .5s and 2s. The tone quality is reminiscent of a plucked string instrument.




```
COMMENT---------------------JCR250---------------------;
COMMENT: SAME SPECTRUM FOR REEDY AND PLUCKED
SOUNDS;
COMMENT:SAMPLING RATE 20 KC;     SIA 0 4 20000;
INS 0 1;SET P8;OSC P5 P7 B3 F2 P30;OSC B3 P6 B3 F1 V1;OUT B3
B1;END;
INS 0 2;SET P8;OSC P5 P7 B3 F2 P30;OSC B3 P6 B3 F1 V2;OUT B3
B1;END;
INS 0 3;SET P8;OSC P5 P7 B3 F2 P30;OSC B3 P6 B3 F1 V3;OUT B3
B1;END;
INS 0 4;ENV P5 F8 B3 P7 P8 P9 P30;OSC B3 P6 B3 F1 P29;OUT B3
B1;END;
COMMENT:TO SET GENERAL CONVT;
SV2 0 10 2 6 -7;
SV2 0 20 2 6 -7;
SV2 0 30 2 6 -7;
SV2 0 40 2 6 107;
GEN 0 2 1 40 30 35 50 10 20 15 0 2 5 3 11;
GEN 0 3 2 0 10 8 6 7 6;
GEN 0 3 3 0 7 8 10 5 5;
GEN 0 1 4 .6 0 .9 120 .7 300 .8 400 .6 512;
GEN 0 1 5 .5 0 .6 240 .5 512;
GEN 0 1 6 .6 0 .9 20 .3 320 0 512;
GEN 0 1 7 .5 0 .8 40 .2 300 0 512;
GEN 0 6 8 9 .99 .99 9;
COMMENT:BREIZ BOMBARDE TYPE;
COMMENT: ONE SINGLE VOICE;
NOT 1    1 .5  600 486 .5;NOT 1.5  1 .25 600 615 .25 4;
NOT 1.75 1 .25 600 648 .25;NOT 2    1 .5  600 729 .5;
NOT 2.5  1 .25 600 972 .25;NOT 2.75 1 .25 600 890 .25;
NOT 3    1 .25 600 820 .25;NOT 3.25 1 .25 600 729;
```

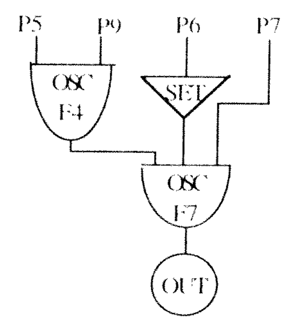
```
NOT 3.5  1 .5  600 820 .5;NOT 4 1 2 600 729 2 6;
SEC 7;
COMMENT:THREE VOICES FOR CHORAL EFFECT;
NOT 1 1 .5  1200 486 .5 2;NOT 1.03 2 .5  500 492 0 3;
NOT 1.08 3 .5  300 473 0 2;
NOT 1.5  1 .25 1200 615 0 4;NOT 1.53 2 .25 500 610 0 5;
NOT 1.58 3 .25 300 629 0 4;
NOT 1.75 1 .25 1200 648;NOT 1.78 2 .25 500 660;
NOT 1.83 3 .25 300 625;
NOT 2   1 .5  1200 729;NOT 2.03 2 .5  500 719;
NOT 2.08 3 .5  300 741;
NOT 2.5  1 .25 1200 972;NOT 2.53 2 .25 500 990;
NOT 2.58 3 .25 300 950;
NOT 2.75 1 .25 1200 890;NOT 2.78 2 .25 500 880;
NOT 2.83 3 .25 300 884;
NOT 3    1 .25 1200 820;NOT 3.03 2 .25 500 830;
NOT 3.08 3 .25 300 809;
NOT 3.25 1 .25 1200 820;NOT 3.28 2 .25 500 835;
NOT 3.33 3 .25 300 807;
NOT 3.5  1 .5  1200 820;NOT 3.53 2 .5  500 848;
NOT 3.58 3 .5  300 800;
NOT 4    1 2   1200 729 0 6;NOT 4.03 2 1.99 500 722 0 7;
NOT 4.08 3 1.92 300 743 0 6;
SEC 8;
COMMENT:PLUCKED SOUND;
NOT 1  4  .5 600 486  .01 0 2;NOT 1.5  4 .25 700 615  .01 0 1;
NOT 1.75 4 .25 700 648  .01 0 1;
NOT 2  4   1 600 486  .01 0 1;NOT 2    4   1 600 615  .01 0 1.5;
NOT 2  4   1 600 729  .01 0 1.5;
NOT 2.5  4 .50 700 1944 .01 0 .9;NOT 2.75 4 .50 700 1728 .01 0 .9;
NOT 3    4 .25 700 1640 .01 0 .5;NOT 3.25 4 .25 700 1458 .01 0 .5;
```



```
NOT 3.5  4 .5  700 1640 .01 0 1;NOT 4    4 .9  600 1458 .01 0 1;
NOT 4    4 .8  600 1230 .01 0 1;NOT 4    4 .9  600 731  .01 0 1;
TER 6;
```

#300 (Track 24)

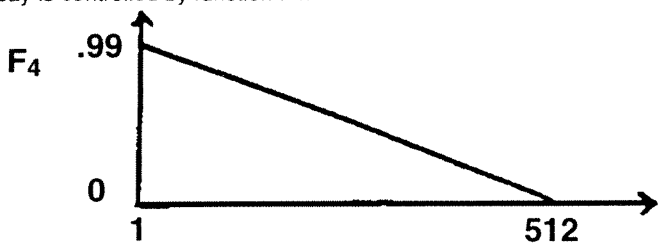
This run compares different decays. Instrument #1 is diagrammed here.



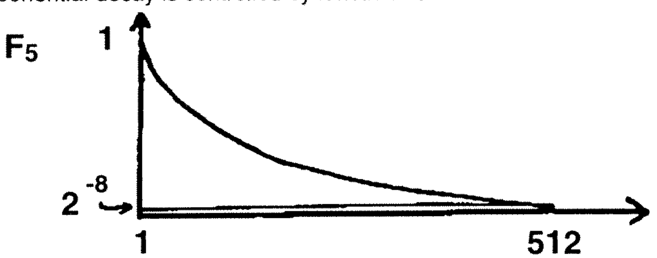
Instrument 1

All 4 instruments are similar, they only differ by the function numbers. (For simultaneous voices, one cannot use the same instrument with different functions; for successive notes one could redefine the function or use SET unit generator).

The first section compares linear and exponential decay. Linear decay is controlled by function F4:



Exponential decay is controlled by function F5:



1st note: linear decay, duration 2s
2nd note: exponential decay, duration 2s
3rd note: linear decay, duration 2s
4th note: exponential decay, duration 4s
(Linear decay seems to decay slowly then suddenly disappears; exponential decay is more even and gives a resonance impression. The beginning of a linear decaying note is comparable with the beginning of an exponentially decaying note of longer duration).

> Note: $2^{-8}=1/256$. To get an uncut exponential decay, one should make sure that the amplitude controlling function decays to a final value not larger than the inverse of the maximum amplitude used, since when the amplitude is smaller than one sample, the sound is lost in the quantizing noise. (E.g., if maximum amplitude is 1500, one should have a function decaying to $2^{-11}=1/2048$).

The following notes consist of 3 waveshapes--F6, F7, F8-- decaying at different rates; in this order:
5) all 3 waveshapes at same frequency 440, longest decay for component with least high frequency content (a "natural" situation, since high frequencies decay faster in pianos, bells. . .);
6) same as previously, except that components have slightly different frequencies 443, 440, 441--to give beats similar to those due to inharmonicity (or bad tuning) in piano sounds;
7) all 3 waveshapes at frequency 440, with the unnatural situation of having component with more high frequency energy decaying slower;
8) same as previously, with component frequencies 443, 440, 441;
9) same a 5)

10) same as previously, with exaggerated differences in component frequencies (448, 444, 440).

```
COMMENT ----------------JCR300-------------------;
COMMENT:DECAY STUDY ON TAPE 2804;
INS 0 1;OSC P5 P9 B3 F4 P30;SET P6;OSC B3 P7 B3 F7 P29;OUT B3
B1;END;
INS 0 2;OSC P5 P9 B3 F5 P30;SET P6;OSC B3 P7 B3 F6 P29;OUT B3
B1;END;
INS 0 3;OSC P5 P9 B3 F5 P30;SET P6;OSC B3 P7 B3 F7 P29;OUT B3
B1;END;
INS 0 4;OSC P5 P9 B3 F5 P30;SET P6;OSC B3 P7 B3 F8 P29;OUT B3
B1;END;
GEN 0 1 4 0.99 1 0 512;
GEN 0 7 5 -8;
GEN 0 3 6 512 0 10 10 10 0 -10 -10 -10 0;
GEN 0 2 7 1 .5 .3 .2 .15 .12 6;
GEN 0 2 8  1 .2 .05 3;
COMMENT: TO SET GENERAL CONVT;
SV2 0 10 2 7 -9;
SV2 0 20 2 7 -9;
SV2 0 30 2 7 -9;
SV2 0 40 2 7 -9;
COMMENT:TWICE LINEAR THEN EXPONENTIAL DECAY  ONE
COMPONENT ONLY;
NOT 1 1 2 1700 7 440 0 2;NOT 4 3 2 1700 7 440 0 2 ;
NOT 67 1 2 1700 7 440 0 2 ;NOT 9 3 4 1700 7 440 0 4;
SEC 15;
COM TRIPLE DECAY;
NOT 1 2 .1 1000 6 440 0 .1;
NOT 1 3 1.8 350 7 440 0 1.8;
```



```
NOT 1 4 3 200 8 440 0 4 ;
SEC 5;
NOT 1 2 .1 1000 6 443 0 .1;
NOT 1 3 1.8 350 7 440 0 1.8;
NOT 1 4 3 200 8 441 0 4 ;
SEC 5;
NOT 1 4 .1 1000 8 440 0 .1;
NOT 1 3 1.8 350 7 440 0 1.8;
NOT 1 2 3 200 6 440 0 4;
SEC 5;
NOT 1 4 .1 1000 8 443 0 .1;
NOT 1 3 1.8 350 7 440 0 1.8;
NOT 1 2 3 200 6 441 0 4;
SEC 5;
NOT 1 2 .1 1000 6 440 0 .1;
NOT 1 3 1.8 350 7 440 0 1.8;
NOT 1 4 3 200 8 440 0 4;
SEC 5;
NOT 1 2 .1 1000 6 448 0 .1;
NOT 1 3 1.8 350 7 440 0 1.8;
NOT 1 4 3 200 8 444 0 4;
TER 5;
```