FM synthesis examples

Basic FM

This example shows J. Chowning's basic FM instruments with three different sets of parameters to create bell, brass and percussion (wood-drum) sounds (see Dodge & Jerse). The orchestra is based on the two-oscillator FM set-up, plus two envelope generators: for creating dynamic spectra (index envelope) and controlling the amplitude. These envelopes are created by using two oscillators reading envelope function tables. The shape of the envelope will be one of the parameters determining the output sound.

The parameters are (function table numbers refer to the numbers used in the score):

Bell
duration (DUR) = 15 seconds
carrier frequency (FC) = 200 Hz, modulator frequency(FM) = 280 Hz (FC:FM = 5:7)
max value for index of modulation (IMAX) = 10
function table (FN) 2 used for both index & amplitude envelopes

Wood-drum
DUR = 0.2 seconds
FC = 80 Hz, FM = 55 Hz
IMAX = 25
FN 3 for amplitude envelope
FN 4 for index envelope

Brass
DUR = 0.6 seconds
FC = 440 Hz, FM = 440 Hz (FC:FM = 1:1)
IMAX = 5
FN 5 for amplitude envelope

The instrument (see Dodge & Jerse for its signal flowchart):

```
instr 1

idur = p3
iamp = p4
ifp  = p5            ; Fc
ifm =  p6            ; Fm
imax = p7            ; max index
iftable1 =  p8       ; time functions
iftable2 =  p9
ifsine  =   1        ; sinewave

kamp   oscil    iamp, 1/idur, iftable1        ; amplitude envelope
kndx   oscil    ifm*imax, 1/idur, iftable2    ; index envelope
amod   oscil    kndx, ifm, ifsine             ; modulator
acar   oscil    kamp, ifp+amod, ifsine        ; carrier
         out    acar
endin
```

The score will have to define **five** function tables: sinewave, bell envelope, wood-drum amp envelope, wood-drum index envelope and brass envelope. We are also asking for the following p-fields (i-statement): p4 (amp), p5 (FC), p6 (FM), p7 (IMAX), p8 (FN1, amp envelope function table) and p9 (FN2, index envelope function table).

First the function tables:

Sinewave

```
f1 0 1024  10  1 ; sine
```

bell exponential decay (GEN 5)

```
f2 0 1024   5  1   686  0.0001 338 0.0001
```

wood-drum amp envelope, also an exponential curve (GEN 5)

```
f3 0 1024   5  .8  204  1   820 0.001
```

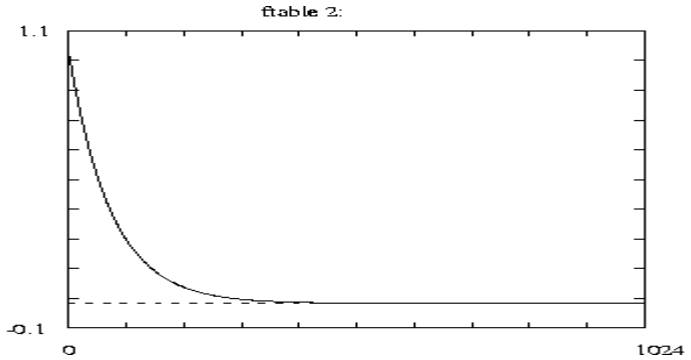
wood-drum index envelope, linear (GEN7)

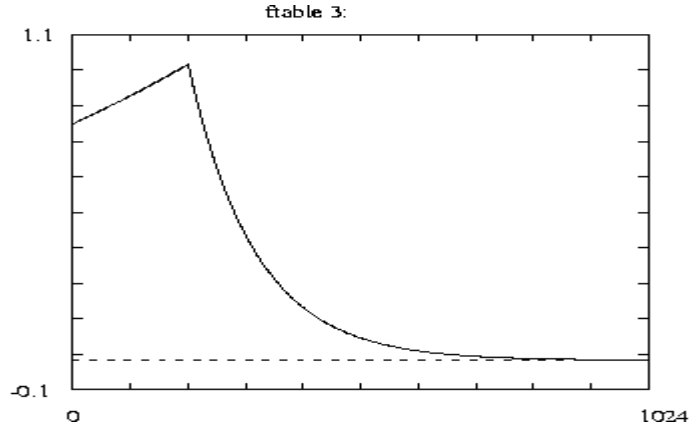
```
f4 0 1024   7  1   156  0      871  0
```

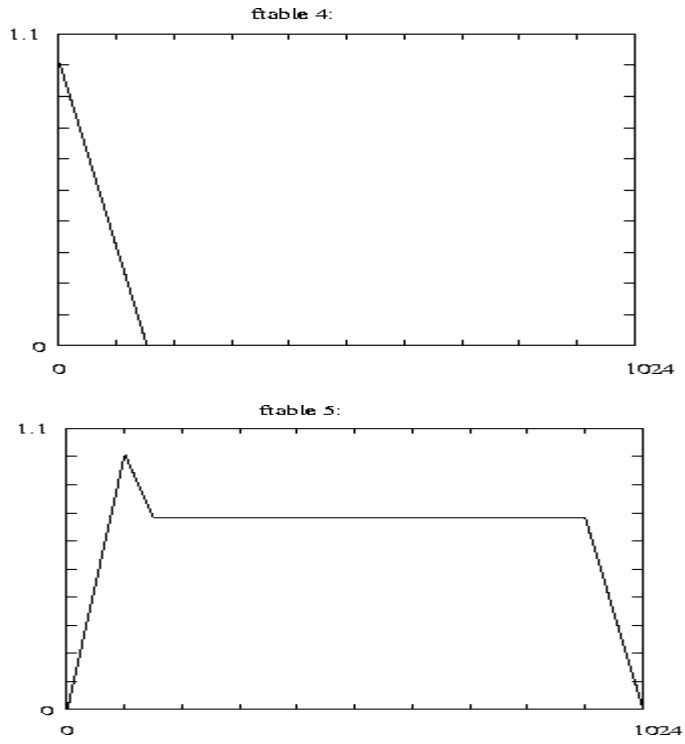
brass envelopes, a simple linear 4-segment envelope

```
f5 0 1024   7  0   102  1  51  .75  768  .75 103  0
```

Here are the plots for the ftables 2 –5:







The following i-statement will generate a bell sound (using the parameters defined above)

```
;    dur amp   fp  fm  imax ftable1 ftable2
i1 0 15  16000 200 280 10   2       2
```

This line will generate a wood-drum sound:

```
;    dur amp   fp  fm  imax ftable1 ftable2
i1 0 .2  16000 80  55  25   3       4
```

A brass-like sound can be generated with the following parameters

```
;    dur amp   fp  fm  imax ftable1 ftable2
i1 0 .6  16000 440 440 5    5       5
```

Using foscil, foscili

The FM pair is such an useful design that csound offers it as a single opcode, named **foscil** (and its interpolating sibling **foscili**):

ar foscil xamp, xcps, xcar, xmod, kndx, ifn

where **xamp, xcps, ifn** are amplitude, base freq and table number (as in oscil) and the extra parameters are: **xcar** and **xmod,** these define the fc:fm ratio of the synthesis, so that fc = xcar*xcps, and fm = xmod * xcps.
kndx, this is the index of modulation.

A Chowning clarinet design using **foscili** would look like this:

```
instr 1

idur=p3
iamp = p4                       ; amp
ifund = cpspch(p5)              ; freq (converted from octave point pitch-class)
imax = p6                       ; max index
imin = 2
isinetab = 1
iamptab = 2
indxtab =  3

kamp  oscil1   0, iamp, idur, iamptab
kndx   oscil1   0, imax-imin, idur, indxtab

aout  foscili   kamp,  ifund,  3, 2,  kndx+imin, isinetab

        out         aout

endin

;score

f1 0 1024 10 1
f2 0 1025  5 0.000001 256 1 640 1 128 0.000001
f3 0 1025  5 1 256 0.000001 768 0.000001


i1  0  .5  10000  8.09  4       ; A
i1  + .5   16000  8.04  5       ; E
i1  +  .5   24000  8.00  6       ; C
```

In this example, I also introduce two new elements of csound programming:

(a) pitch conversion: **cpspch(.)** converts octave point pitch class into Hz
octave point pitch class is writen **oct.pc** (8 is central C octave, C3, and pitch classes are C = 00, C#=01, D=02, D#=03, E=04, ..., A#=10,B=11.

6.07 is a G1
10.04 is E5

(b) The + sign on **p2** indicates that the start time is the sum of the previous start time (p2) plus its duration (p3). It only works on lines with same p1 (instrument number). The start times of the example above will be 0,2,4.