Capitolo 1

Descrizione formale dell'informazione musicale

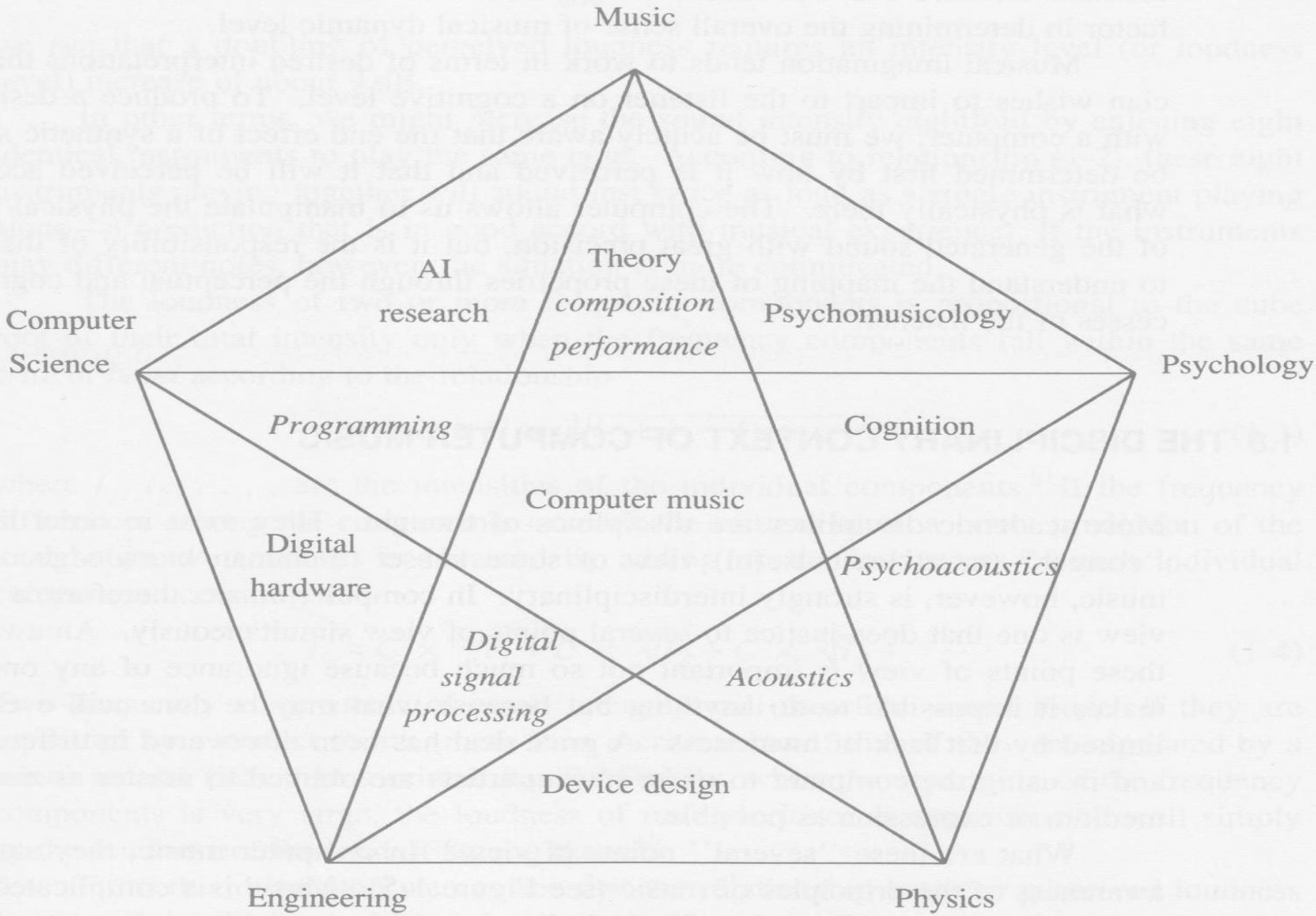


Figure 1-5 Disciplinary context of computer music (essential subdisciplines are shown in *italics*).

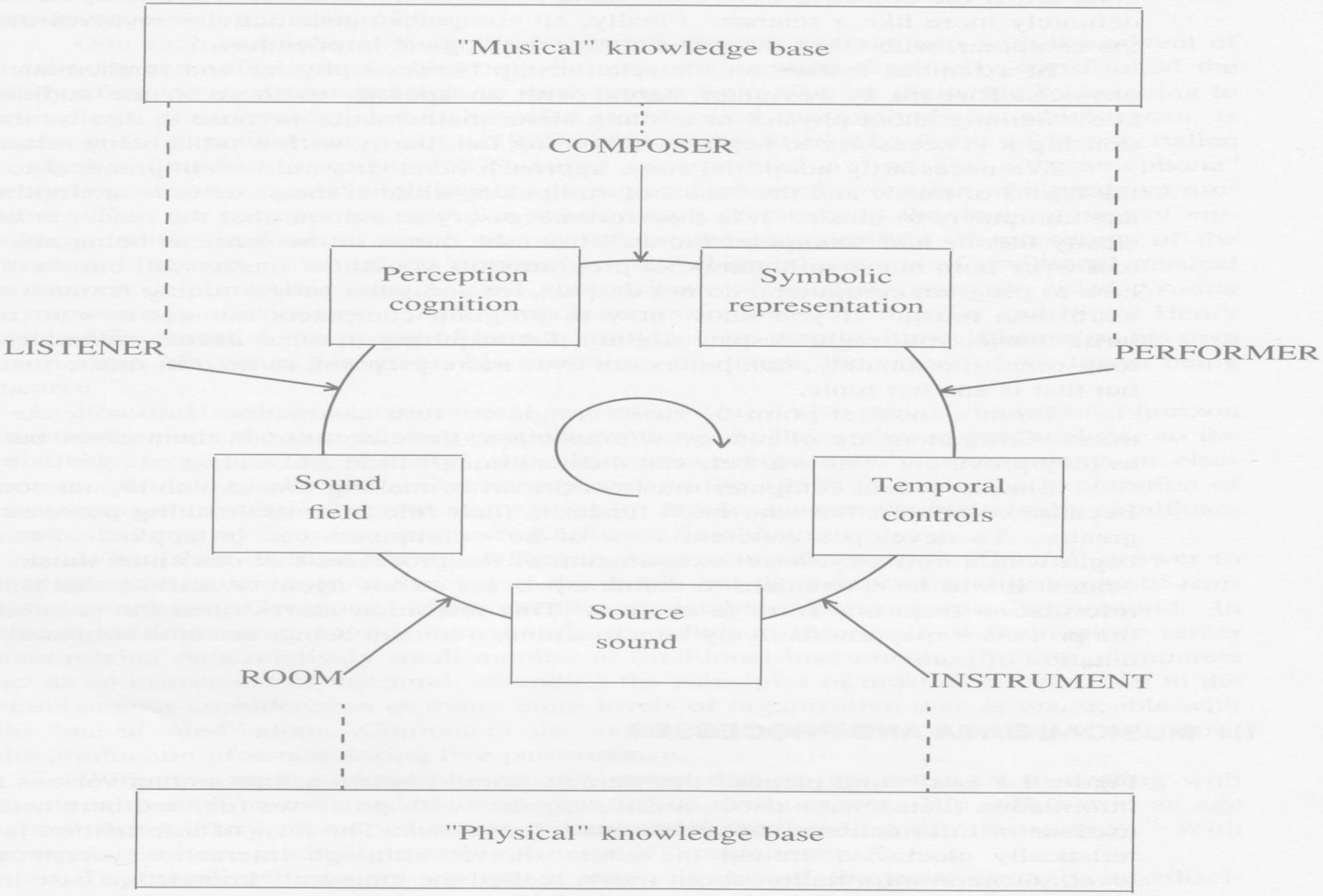


Figure 1-1 Musical data (in boxes) and processors (in capital letters). Data is information that may exist in many forms; processors may be human beings, machines, or a combination of both.

Capitolo 2

Livello del segnale audio

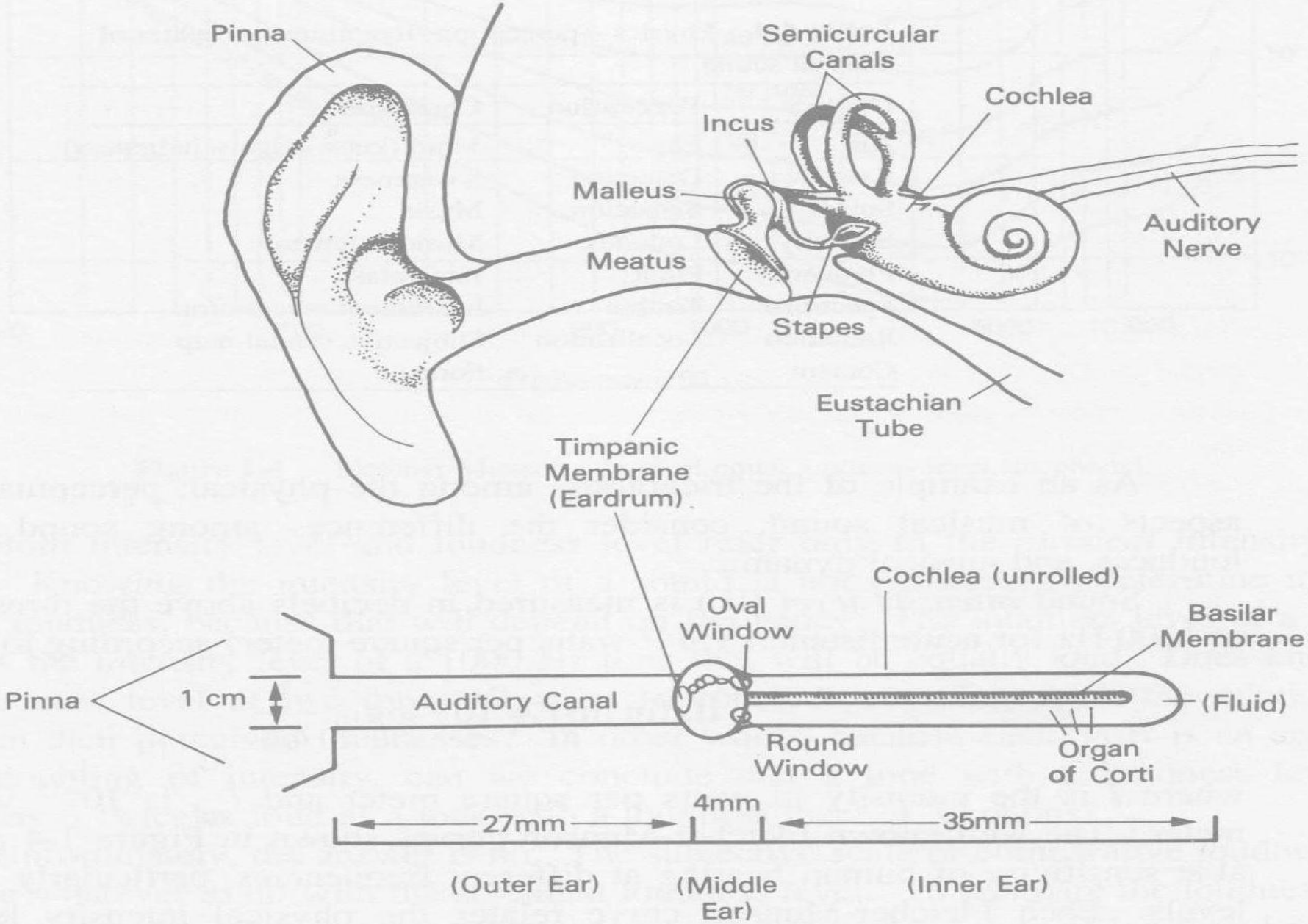


Figure 1-3 The ear.

Table 1-1 Physics→perception→cognition correlates of musical sound

Physics	Perception	Cognition
Air	Ear	Mind (knowledge + judgment)
Existence	Detection	Awareness
Sound	Sensation	Music
Intensity	Loudness	Musical dynamic
Frequency	Pitch	Pitch class
Spectrum	Timbre	Instrument recognition
Radiation	Localization	Subjective spatial map
Content	→	Form

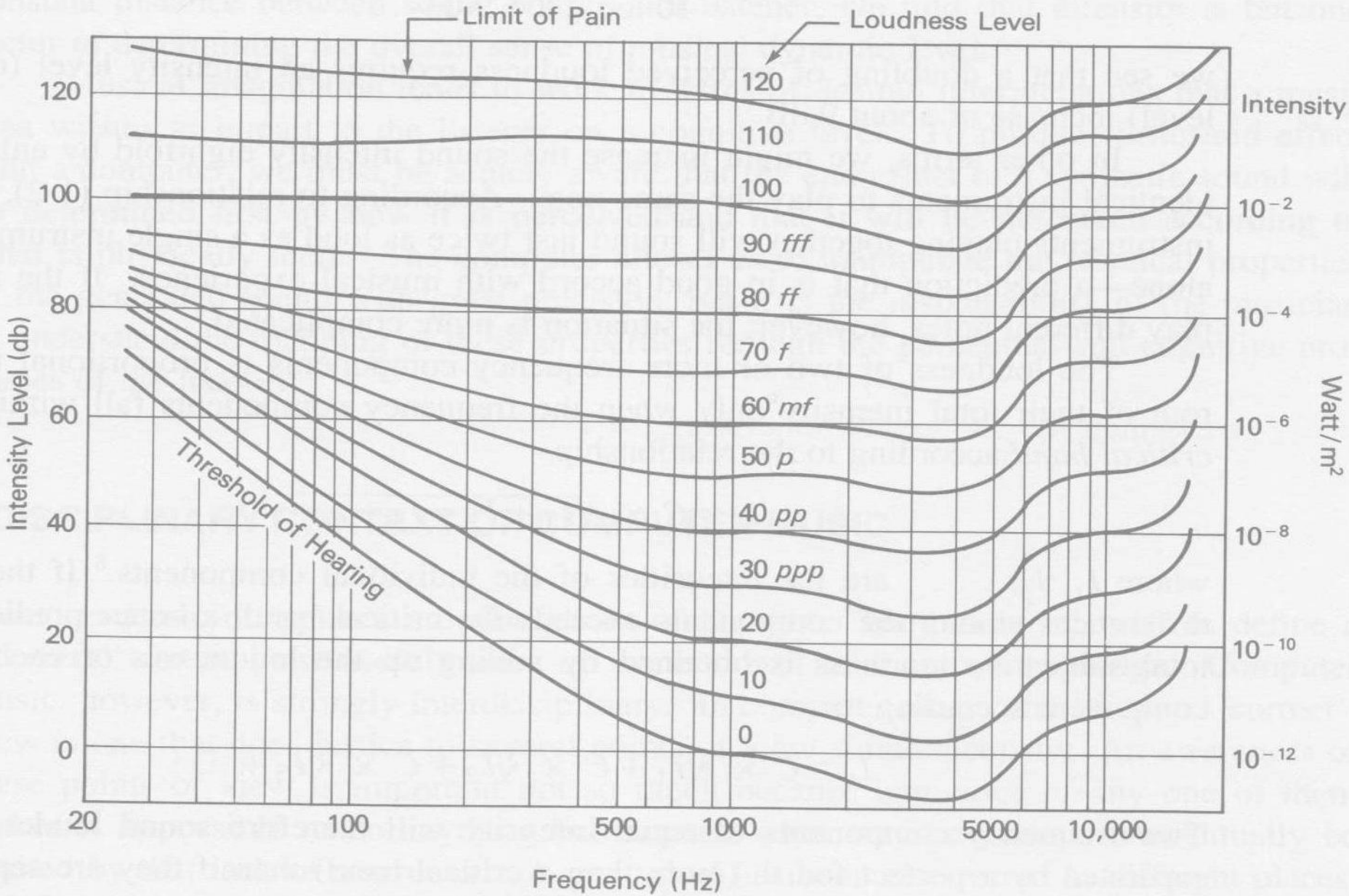


Figure 1-4 Fletcher-Munson curves of equal loudness level (isophons).

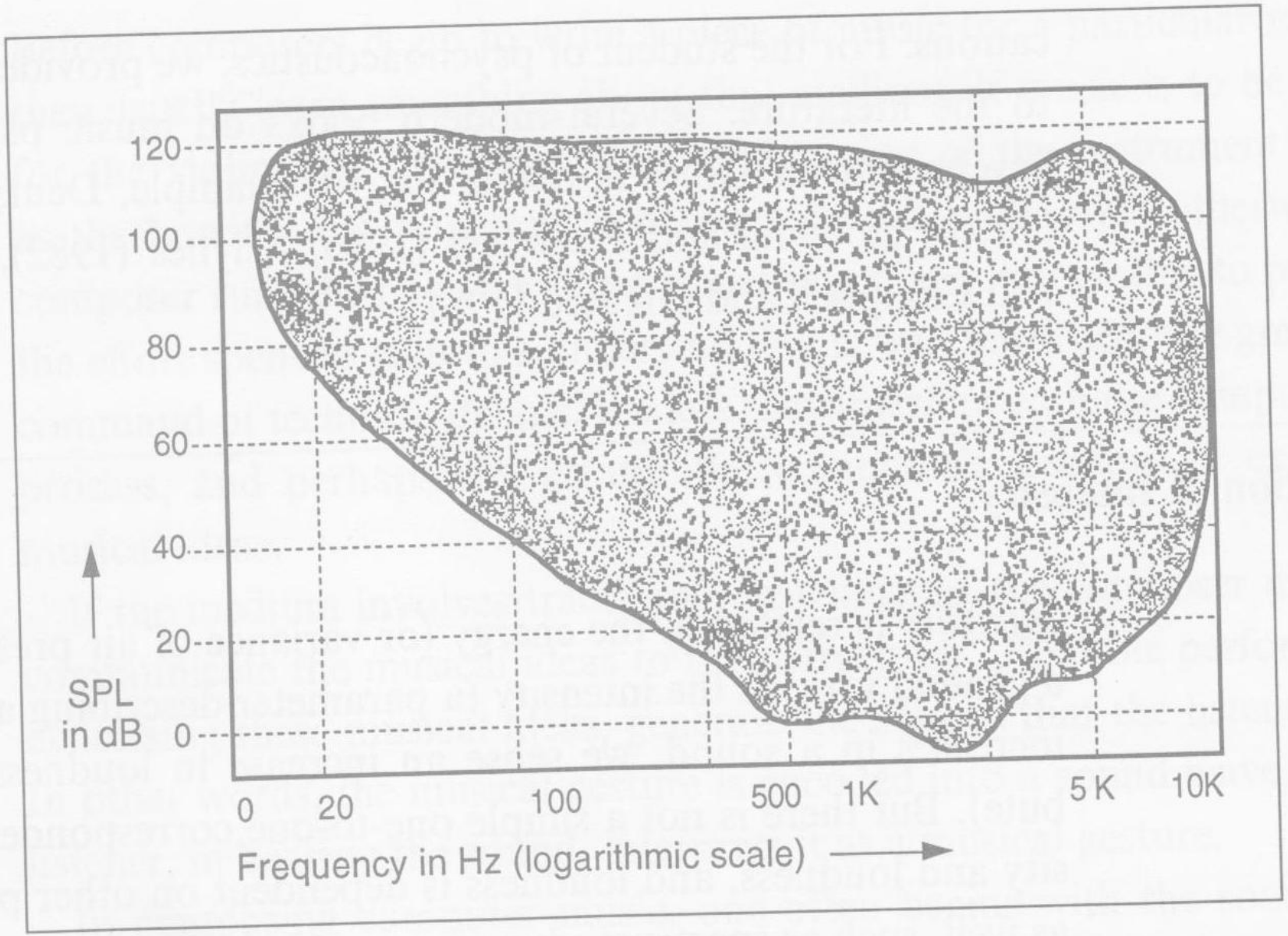


Figure 23.1 The general area of human hearing as a function of frequency and intensity, after data in Winckel (1967).

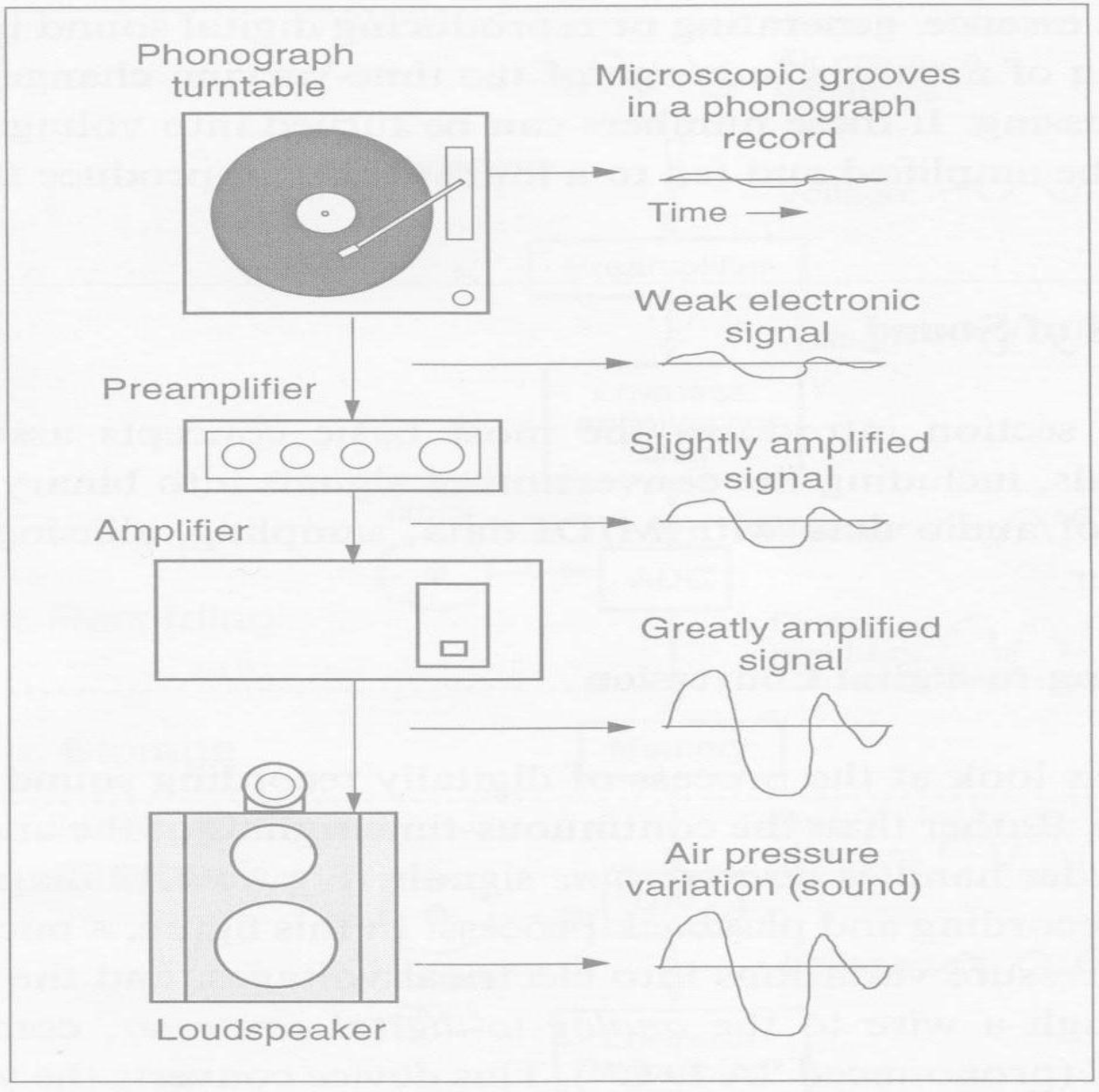


Figure 1.12 The analog audio chain, starting from an analog waveform transduced from the grooves of a phonograph record to a voltage sent to a preamplifier, amplifier, loudspeaker, and projected into the air.

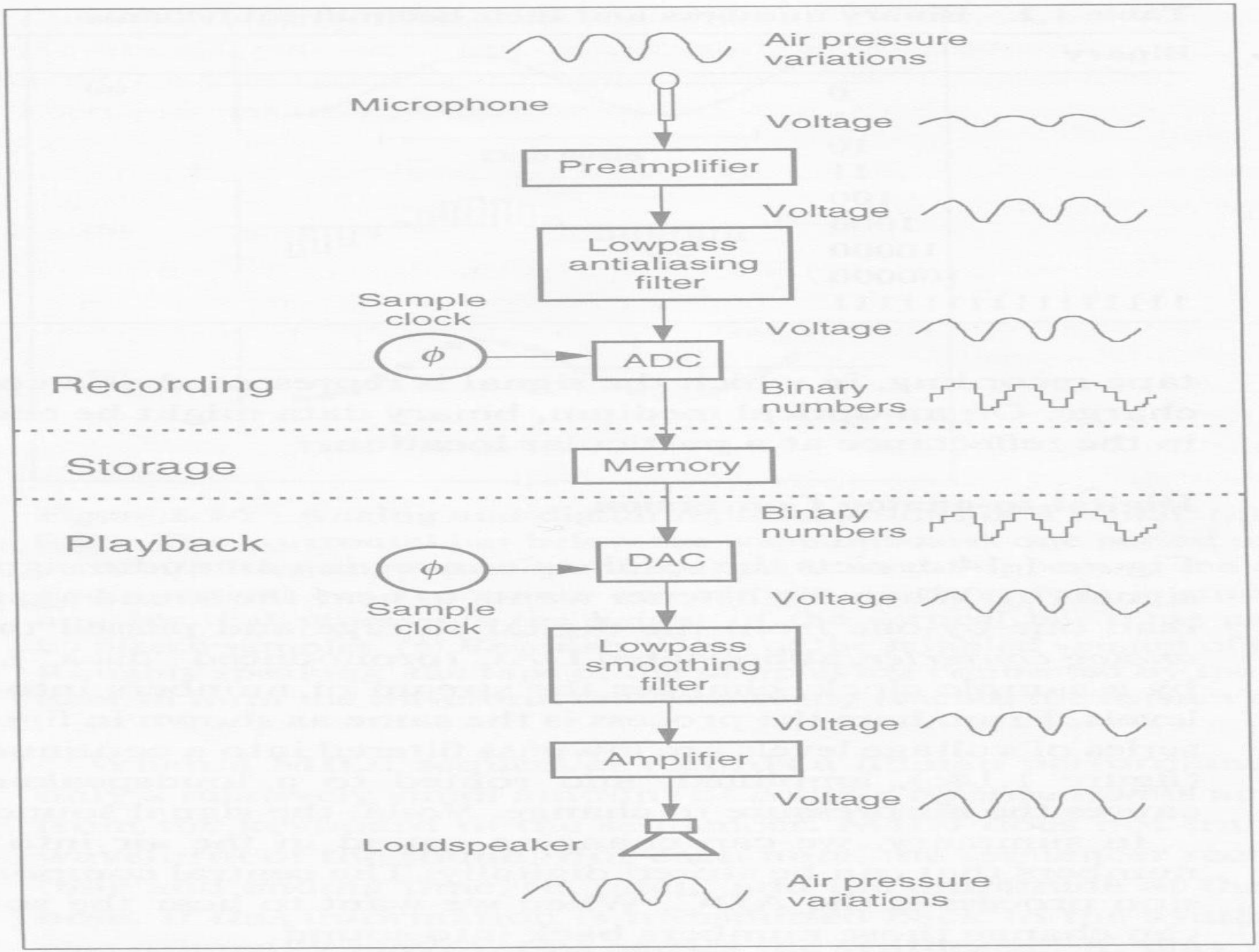


Figure 1.13 Overview of digital recording and playback.

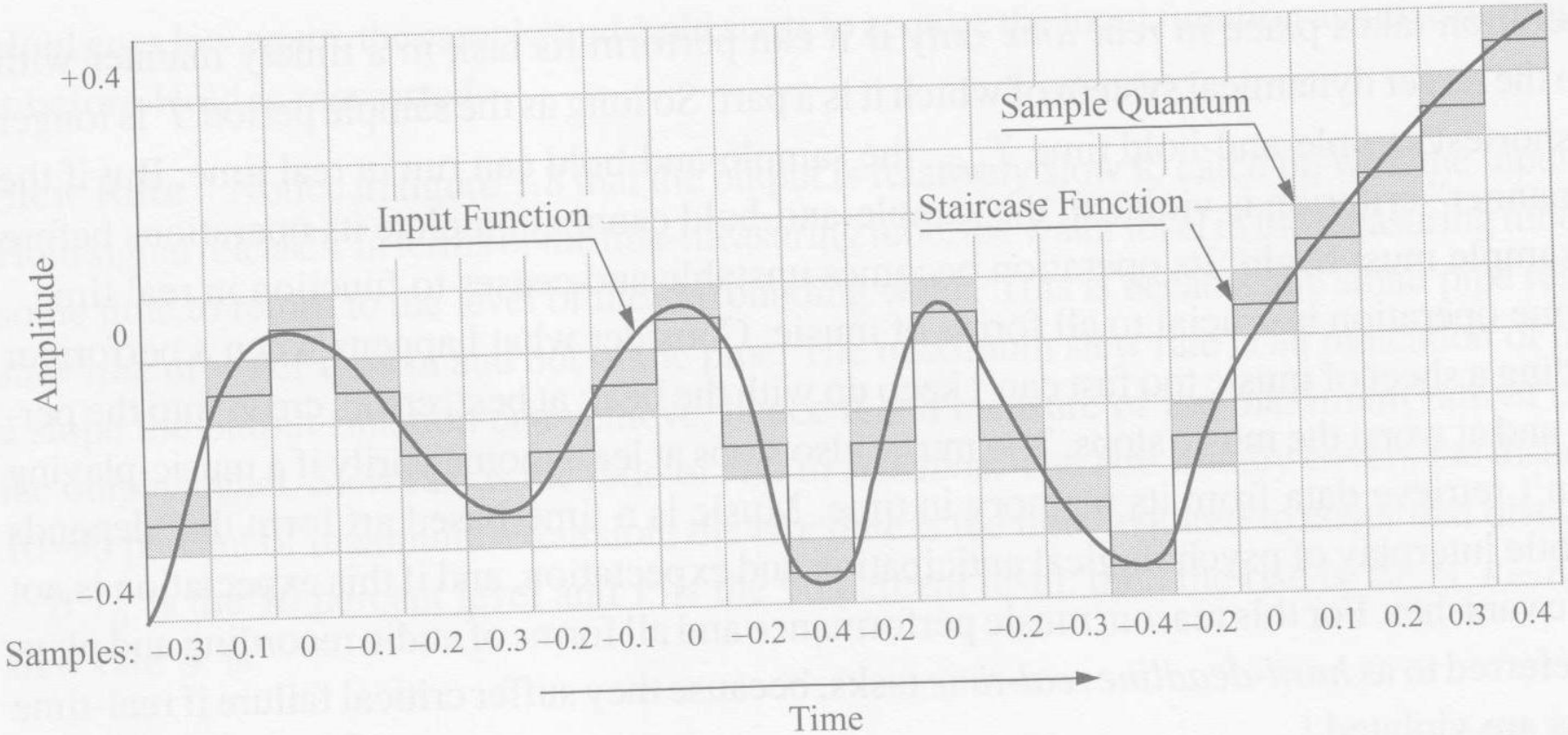


Figure 1.9
Quantization of a signal in time and amplitude.

TABLE 2-1 Linear quantization of a ±10-volt signal into 4 bits. If the voltage level of a sample falls within the minimum and maximum values of a particular region, the binary code associated with that region will be generated. In linear quantization, each region has the same size (1.25 volts in this example).

Quantization Region	Voltage Minimum	Voltage Maximum	Possible Binary Code
1	–10.00	–8.75	0000
2	–8.75	–7.50	0001
3	–7.50	–6.25	0010
4	–6.25	–5.00	0011
5	–5.00	–3.75	0100
6	–3.75	–2.50	0101
7	–2.50	–1.25	0110
8	–1.25	0.00	0111
9	0.00	1.25	1000
10	1.25	2.50	1001
11	2.50	3.75	1010
12	3.75	5.00	1011
13	5.00	6.25	1100
14	6.25	7.50	1101
15	7.50	8.75	1110
16	8.75	10.00	1111

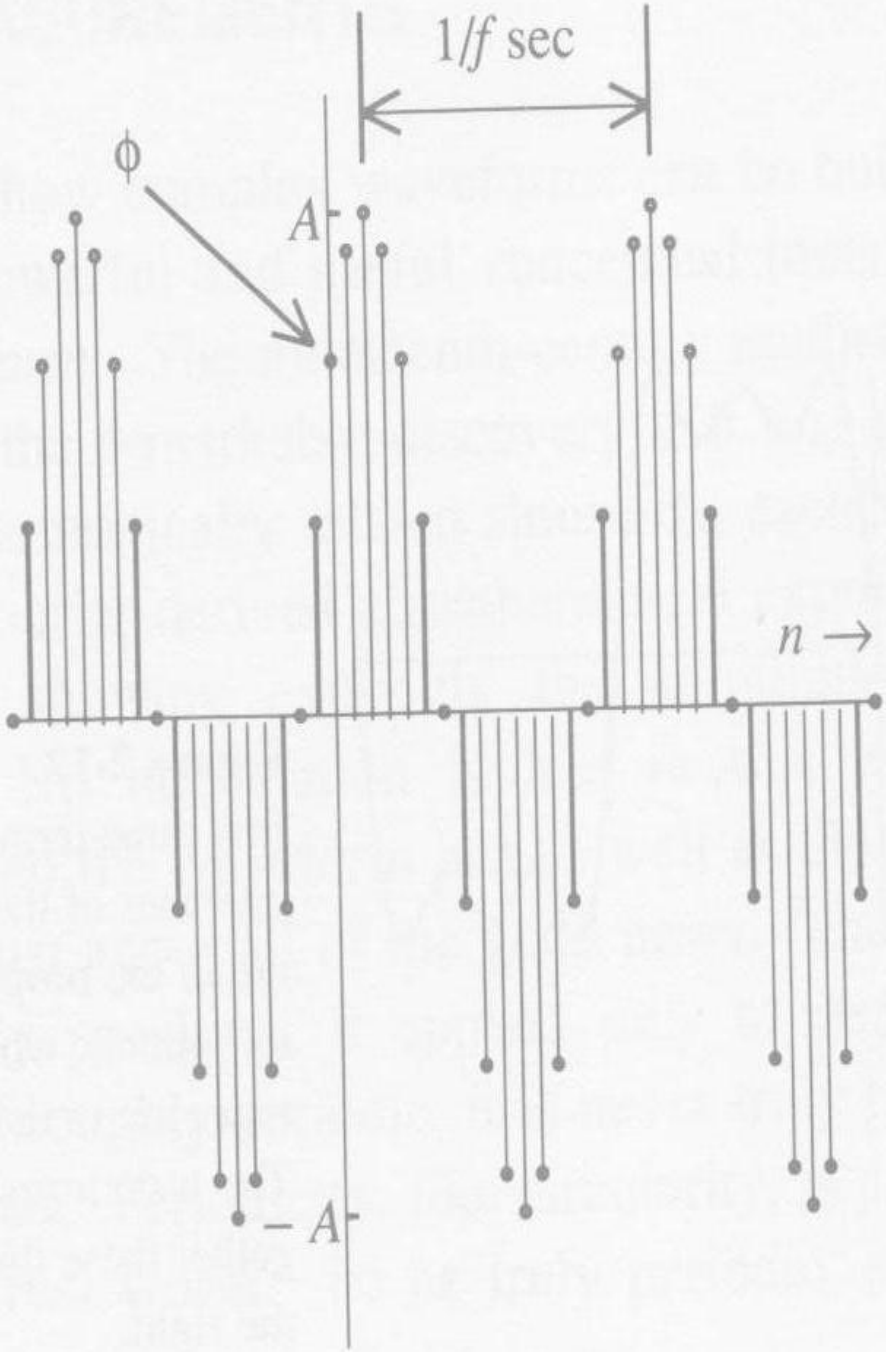


Figure 2-11 Graphical representation of the digital signal $x(n) = A \sin(2\pi f n + \phi)$. Peak amplitude ($A$), frequency ($f$), and phase ($\phi$) are shown as in Figure 2-10. Discrete time (n) is plotted along the horizontal axis, increasing toward the right. The relationship between sample index n and time depends on the sample rate (R) chosen—although in this example it is evident that $R = 16f$ because each period of $x(n)$ is sampled 16 times. Each increment in the value of n represents a time $T = 1/R$ seconds.

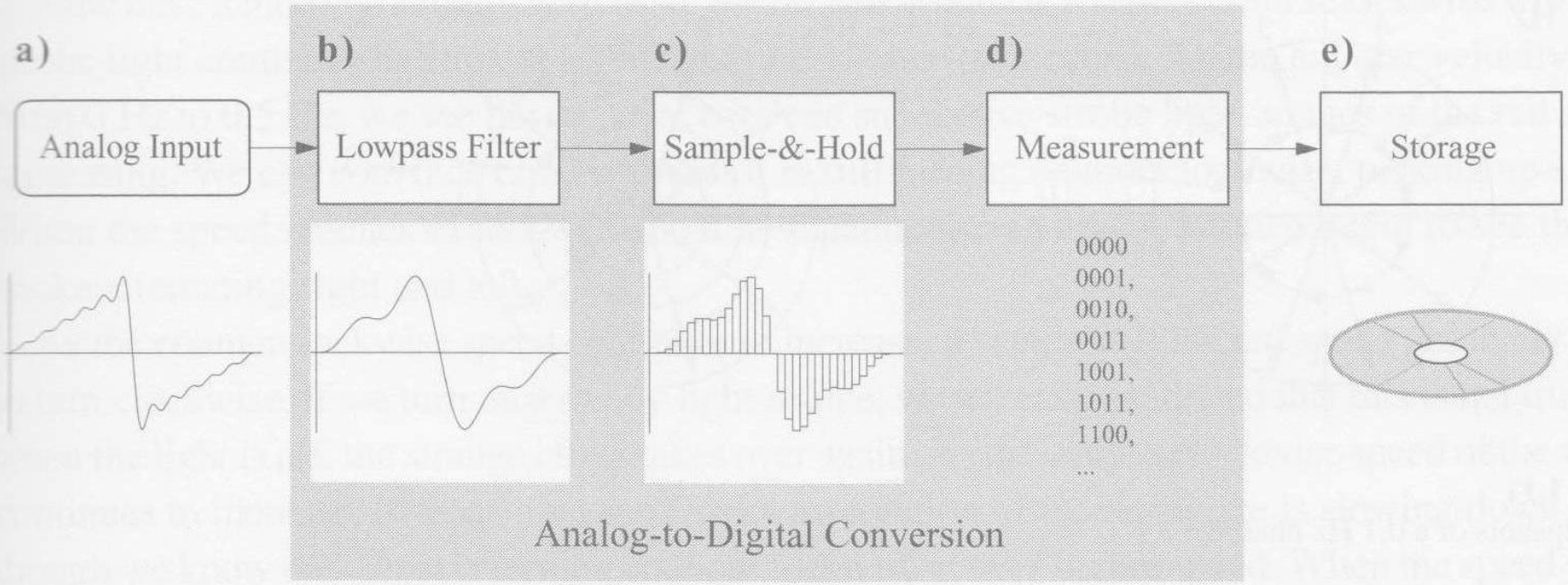


Figure 1.10
Analog-to-digital conversion.

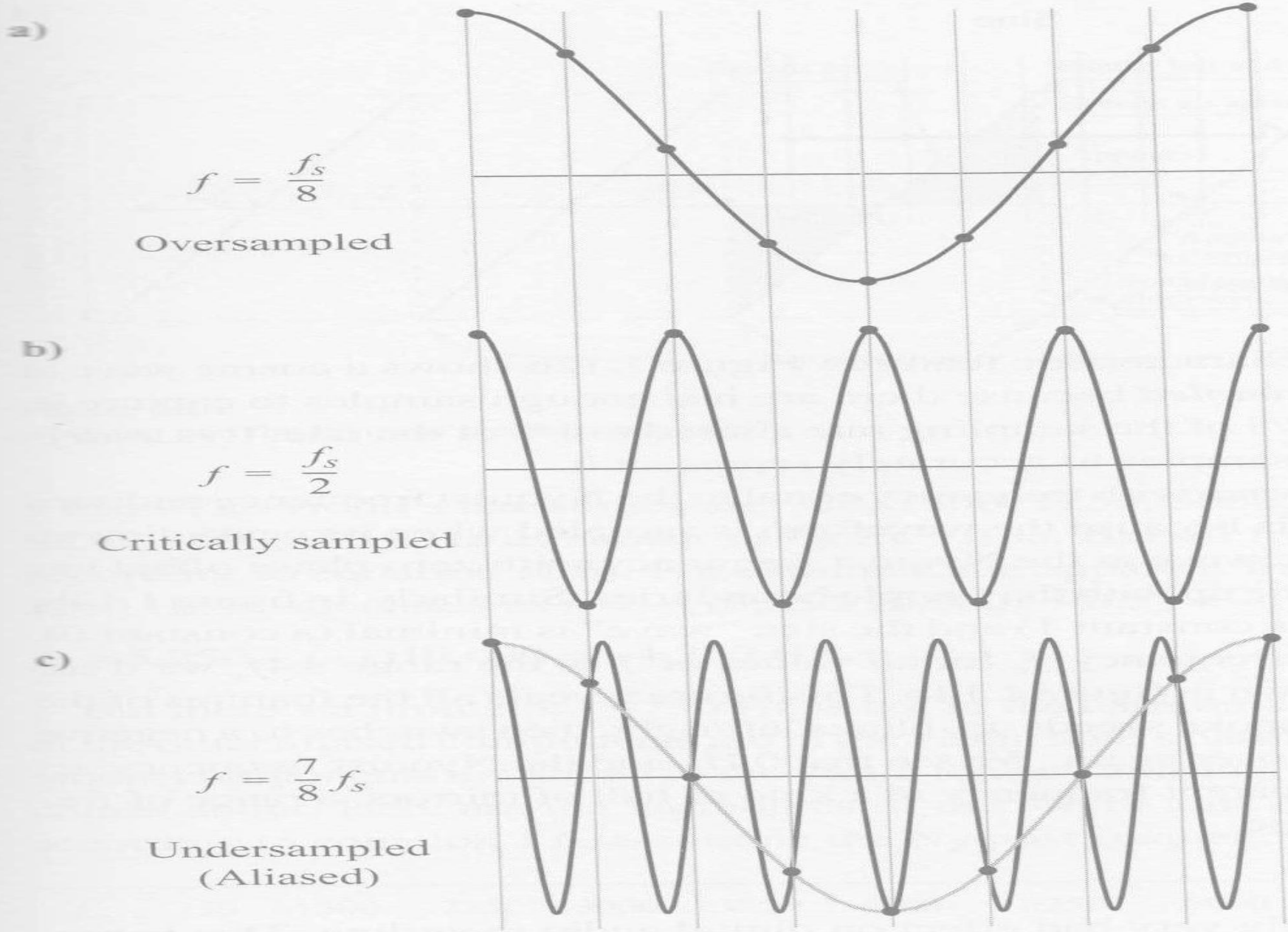


Figure 1.12
Aliasing.

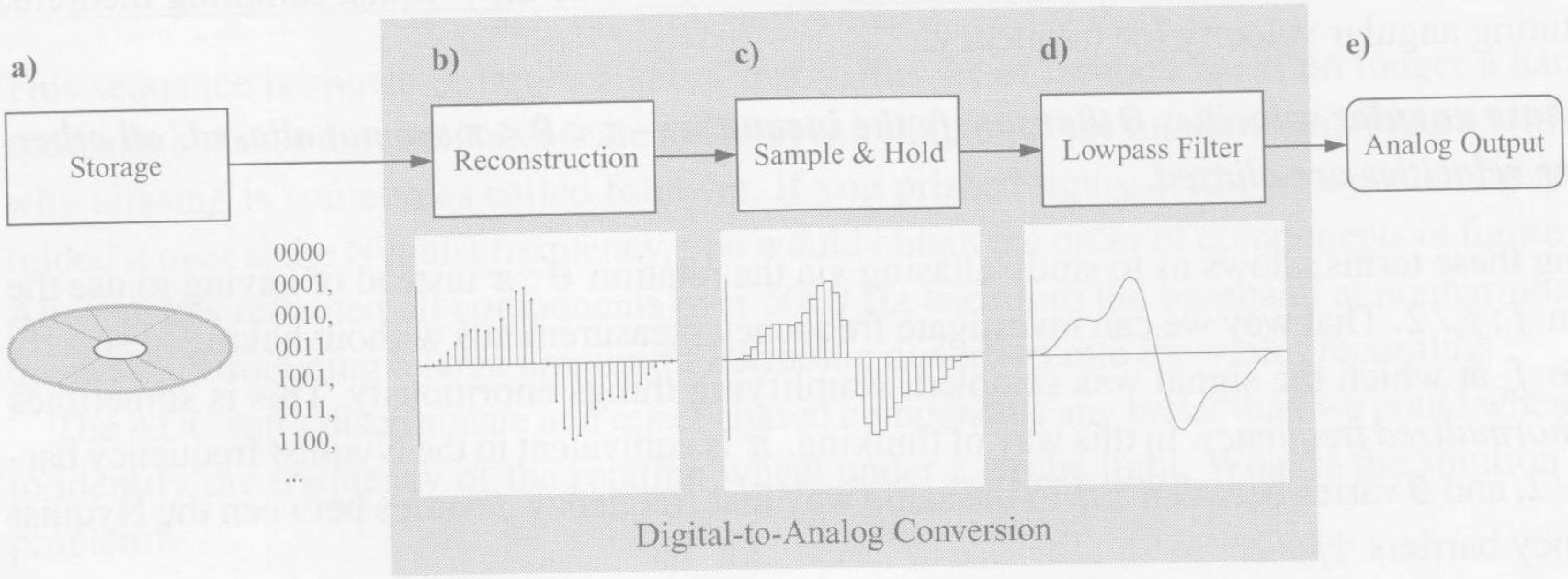


Figure 1.16
Digital-to-analog conversion.

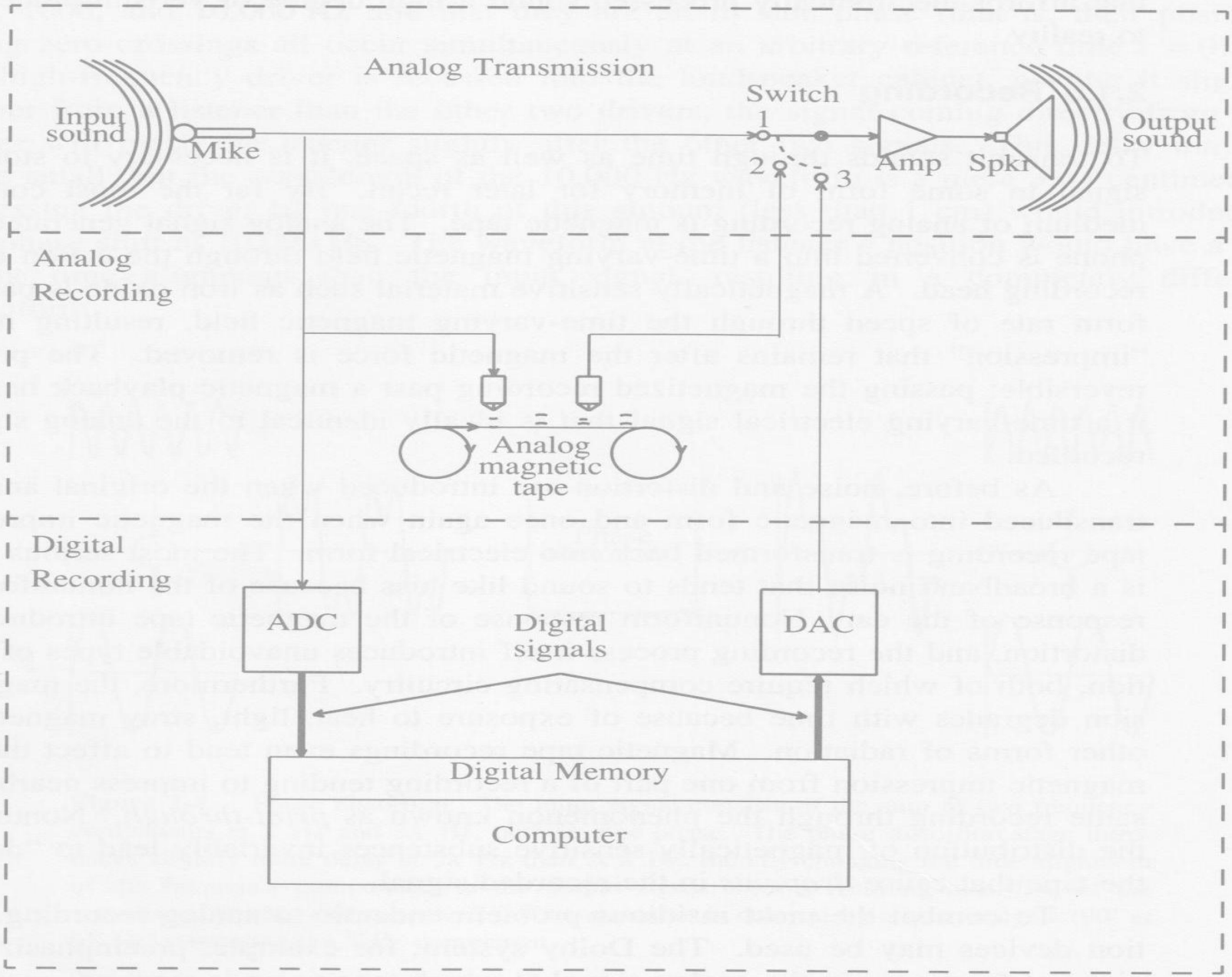


Figure 2-5 Audio signal relationships.

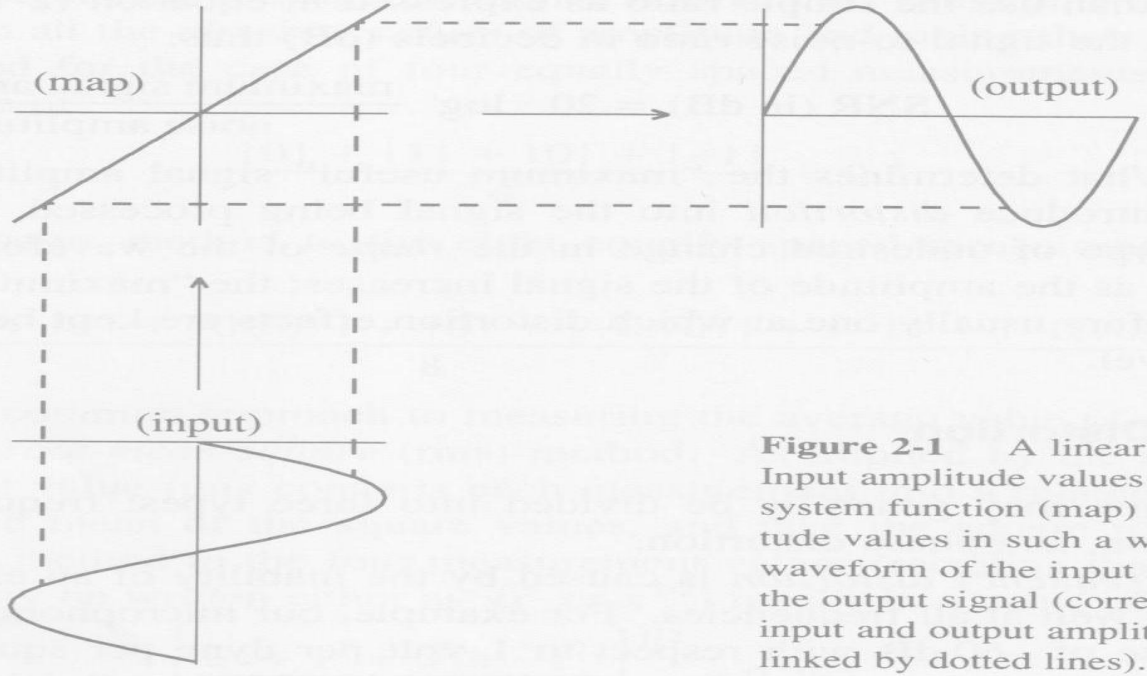


Figure 2-1 A linear system function. Input amplitude values are mapped by the system function (map) into output amplitude values in such a way that the waveform of the input signal is preserved in the output signal (corresponding values of input and output amplitude are shown linked by dotted lines).

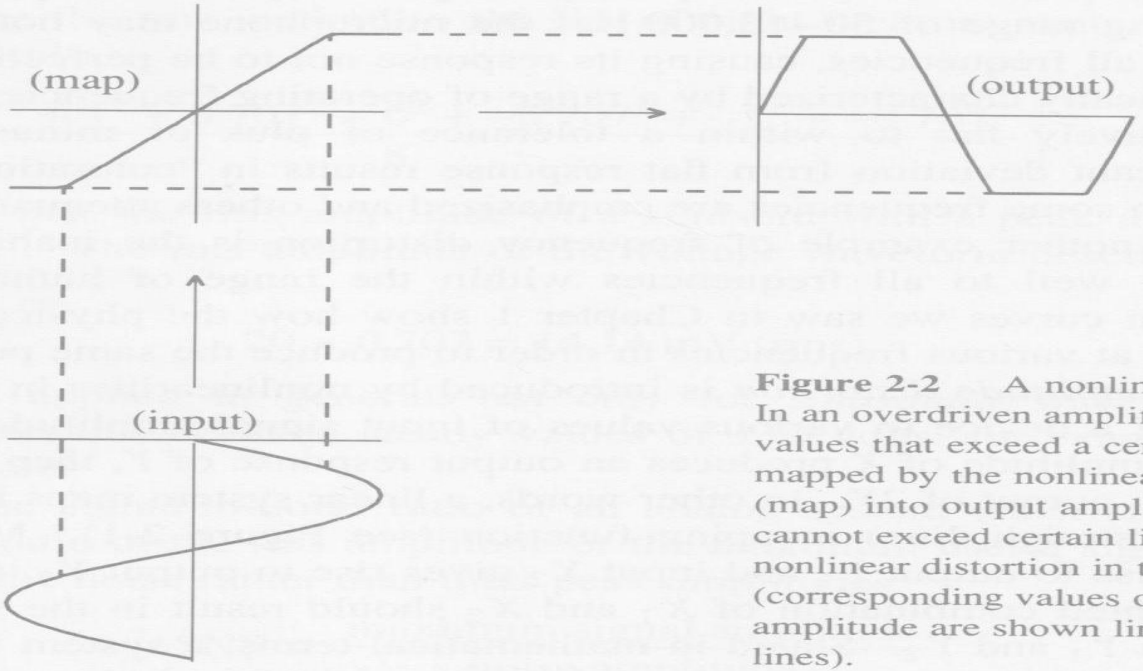


Figure 2-2 A nonlinear system function. In an overdriven amplifier, input amplitude values that exceed a certain value are mapped by the nonlinear system function (map) into output amplitude values that cannot exceed certain limits. This results in nonlinear distortion in the output waveform (corresponding values of input and output amplitude are shown linked by dotted lines).

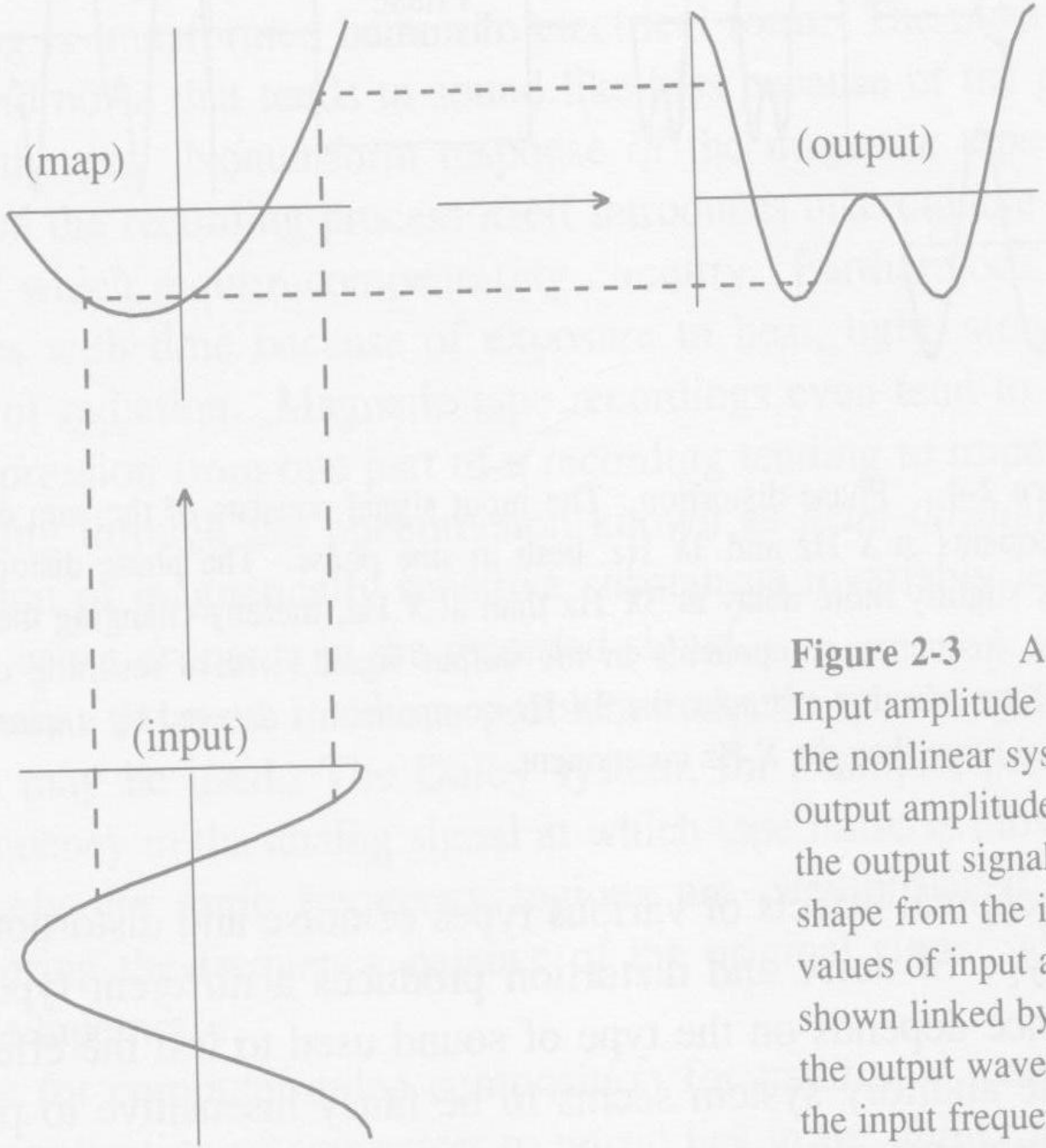


Figure 2-3 A nonlinear system function. Input amplitude values are mapped by the the nonlinear system function (map) into output amplitude values in such a way that the output signal has a completely different shape from the input signal (corresponding values of input and output amplitude are shown linked by dotted lines). In this case the output waveform contains harmonics of the input frequency (harmonic distortion).

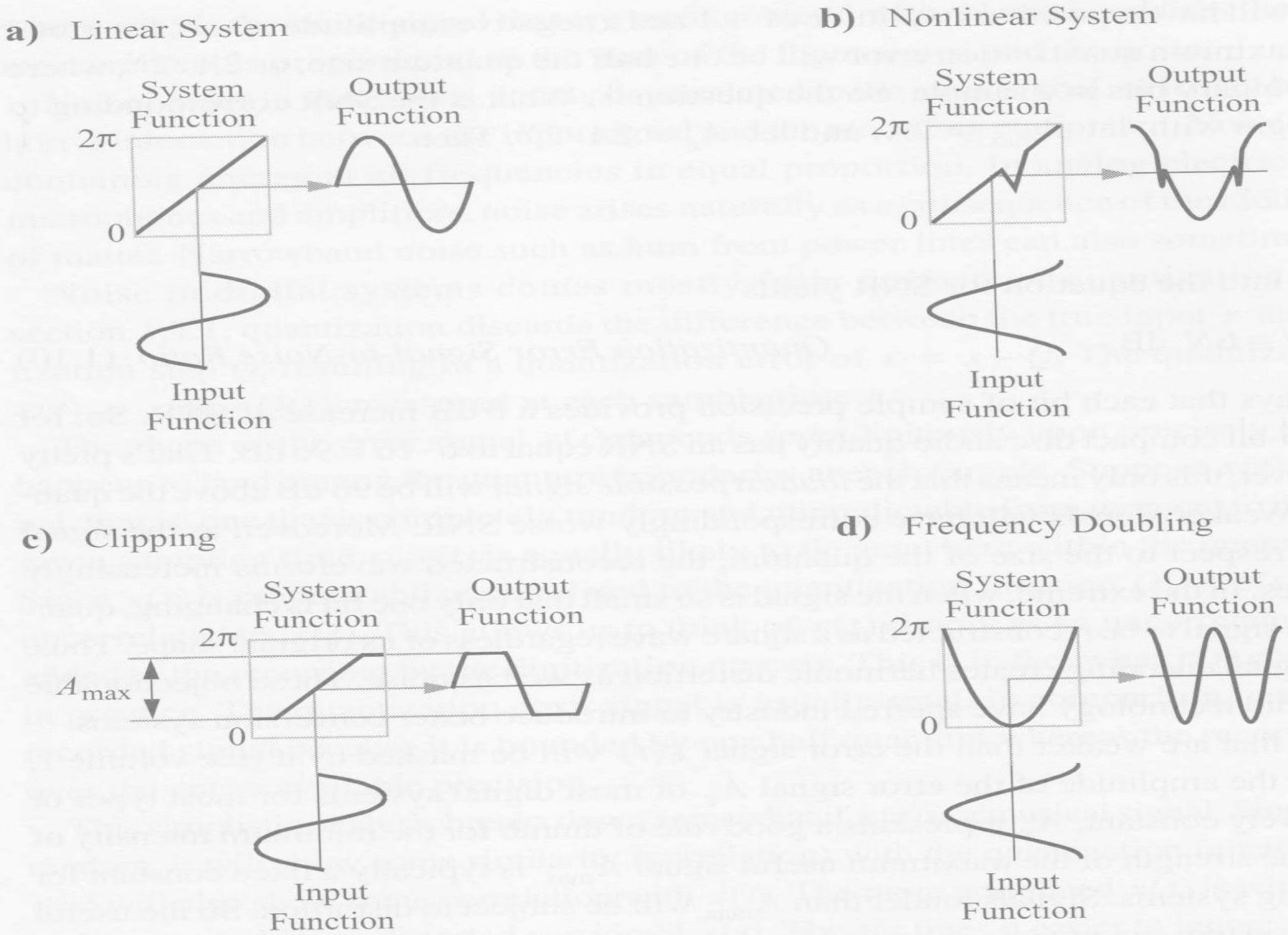


Figure 1.28
Linear and nonlinear system functions.

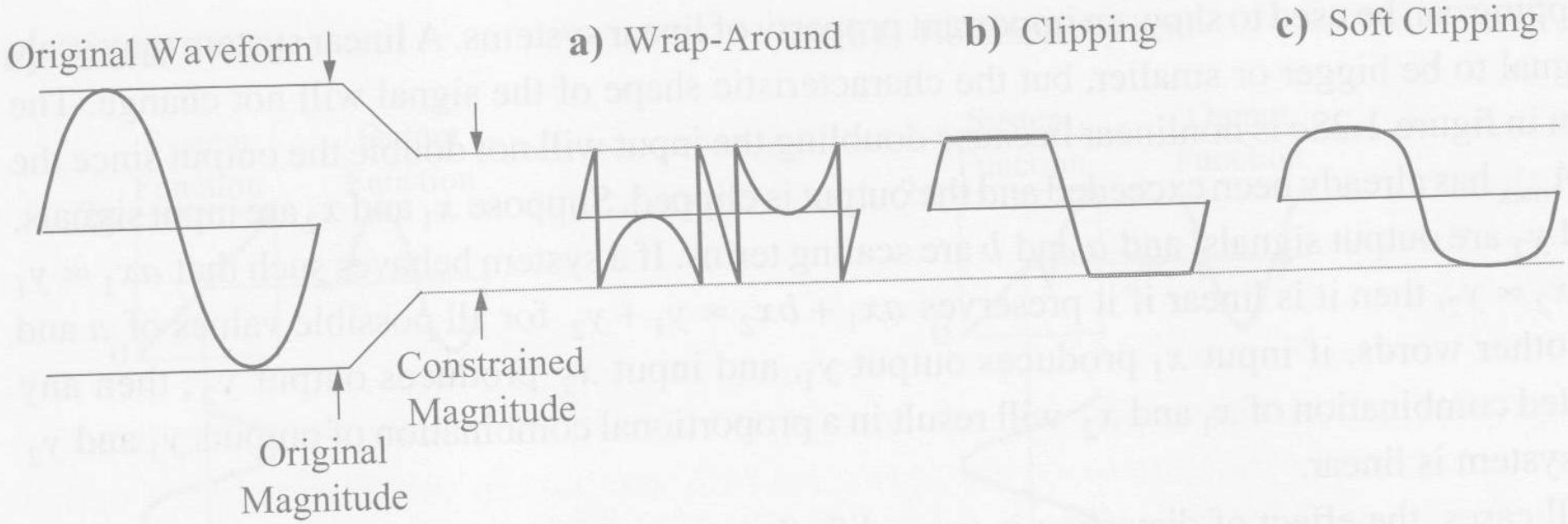


Figure 1.29
Managing overflow.

$$\sigma(x) = \begin{cases} A_{\max}, & x \geq A_{\max}, \\ x, & A_{\max} > x > -A_{\max}, \\ -A_{\max}, & x \leq -A_{\max}. \end{cases}$$

Saturation

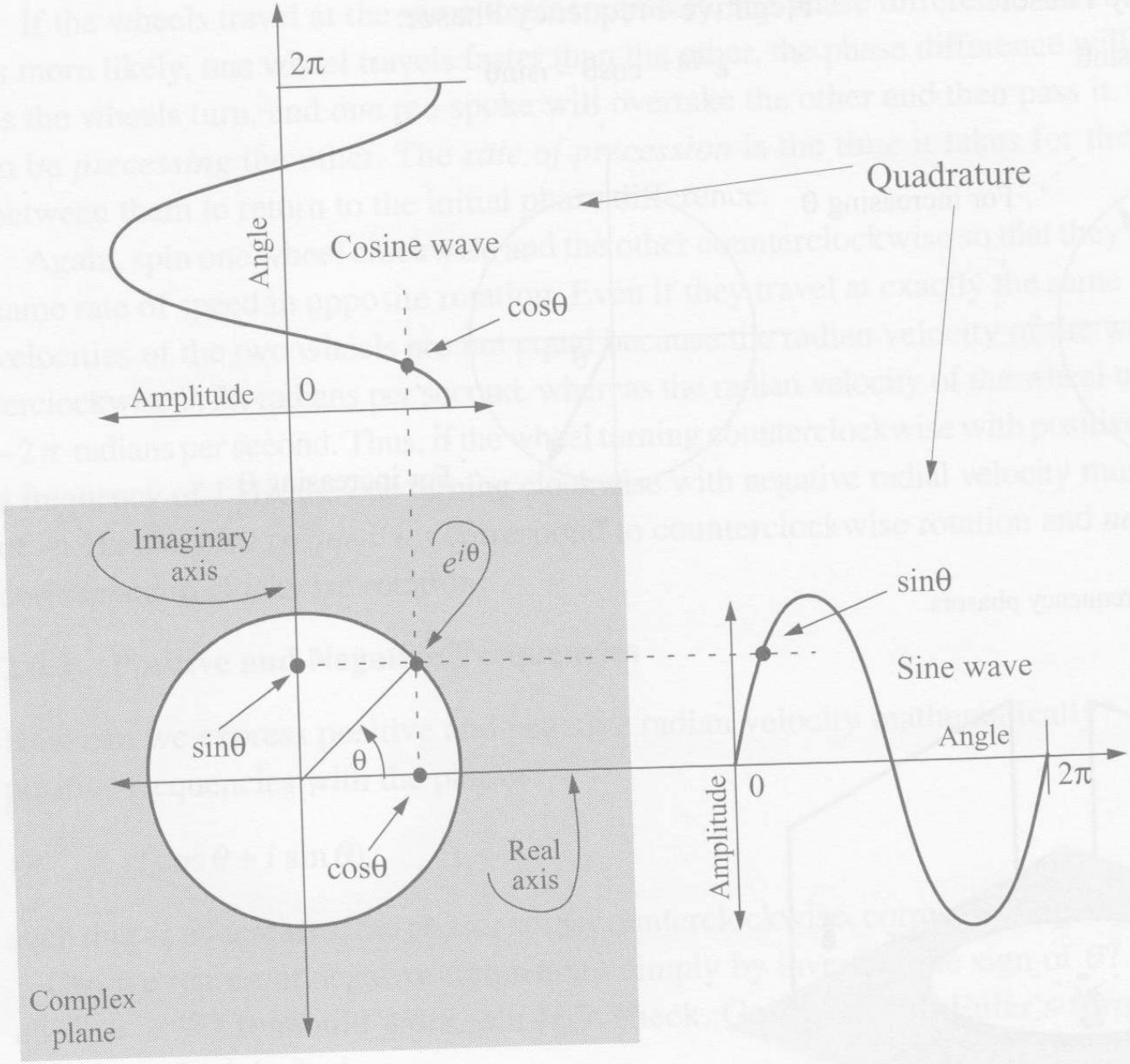


Figure 2.13
Complex phasor projected on the real axis and the imaginary axis.

Capitolo 3

Programmazione timbrica

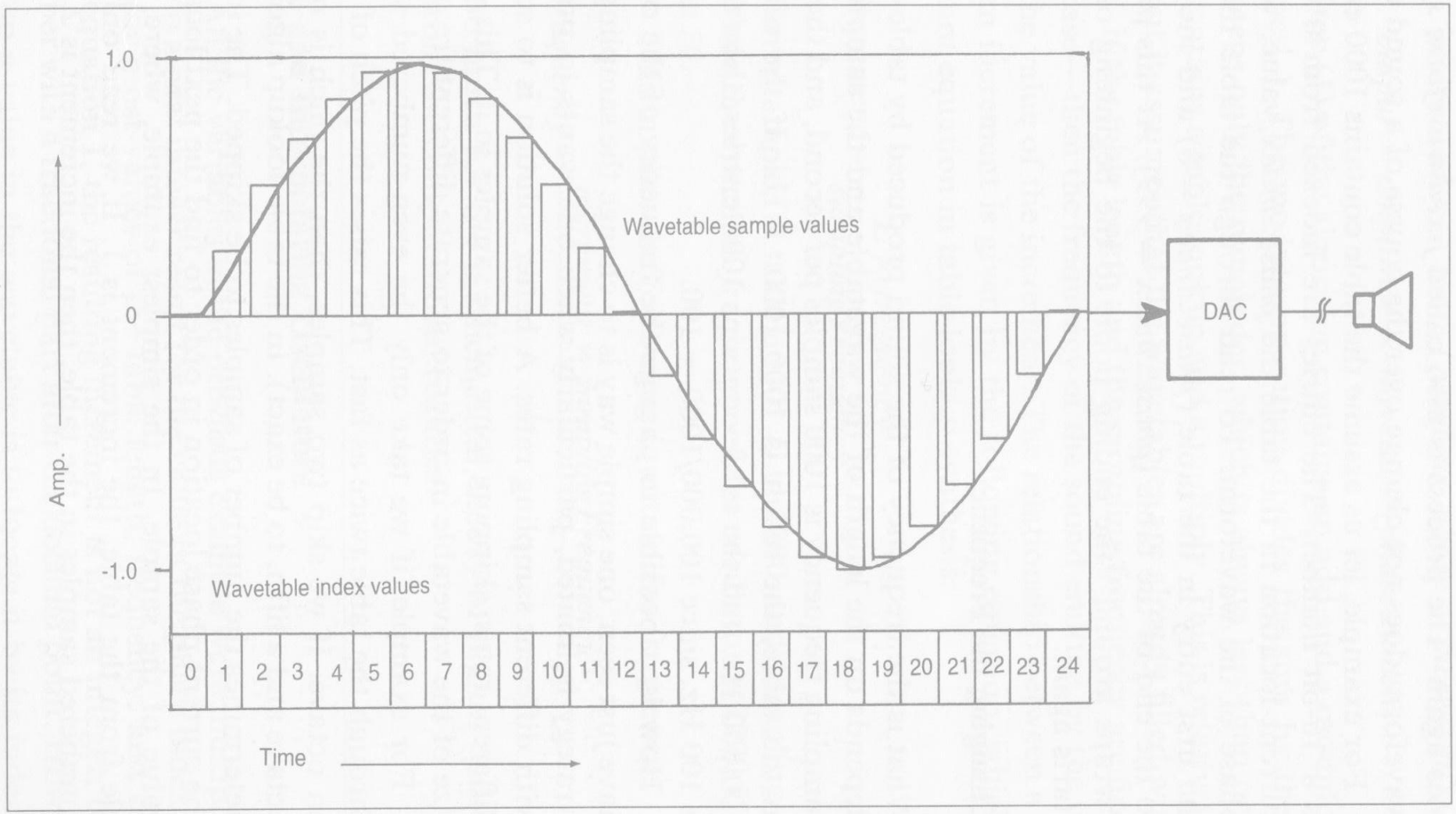


Figure 3.3 Graphical depiction of wavetable-lookup synthesis. The list 0–24 in the lower portion are numbered locations or "table index values." An audio sample value is stored in memory for each index point. The samples are depicted as the rectangles outlining a sine wave in the top portion. For example, Wavetable[0] = 0, and Wavetable[6] = 1. To synthesize the sine wave the computer looks up the sample values stored in successive index locations and sends them to a DAC, looping through the table repetitively.

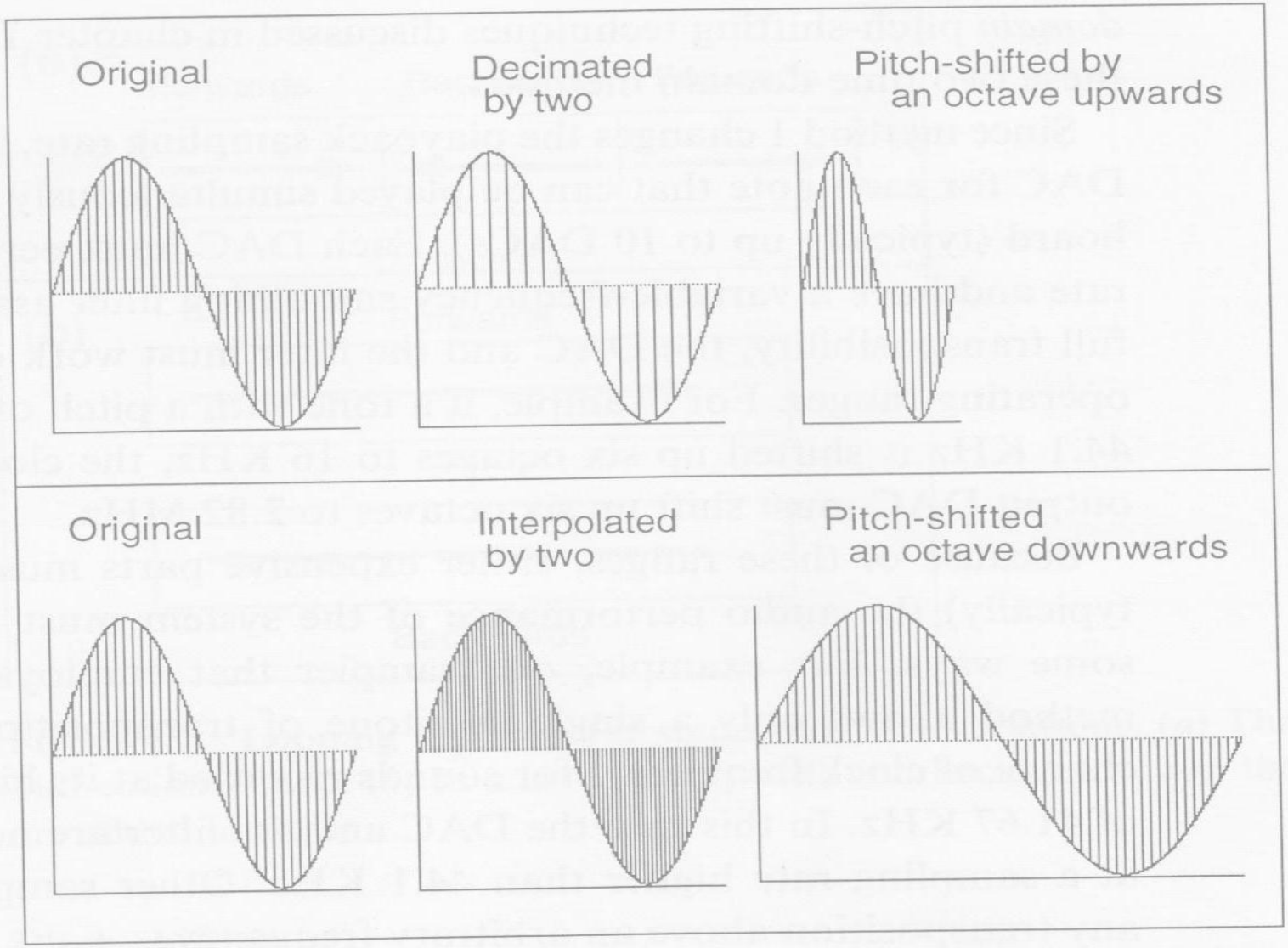


Figure 4.8 Pitch-shifting by sample-rate conversion with a constant playback sampling frequency. (Top) If every other sample is skipped on playback, the signal is decimated and the pitch is shifted up an octave. (Bottom) If twice the number of samples are used by means of interpolation on playback, the signal is shifted down an octave.

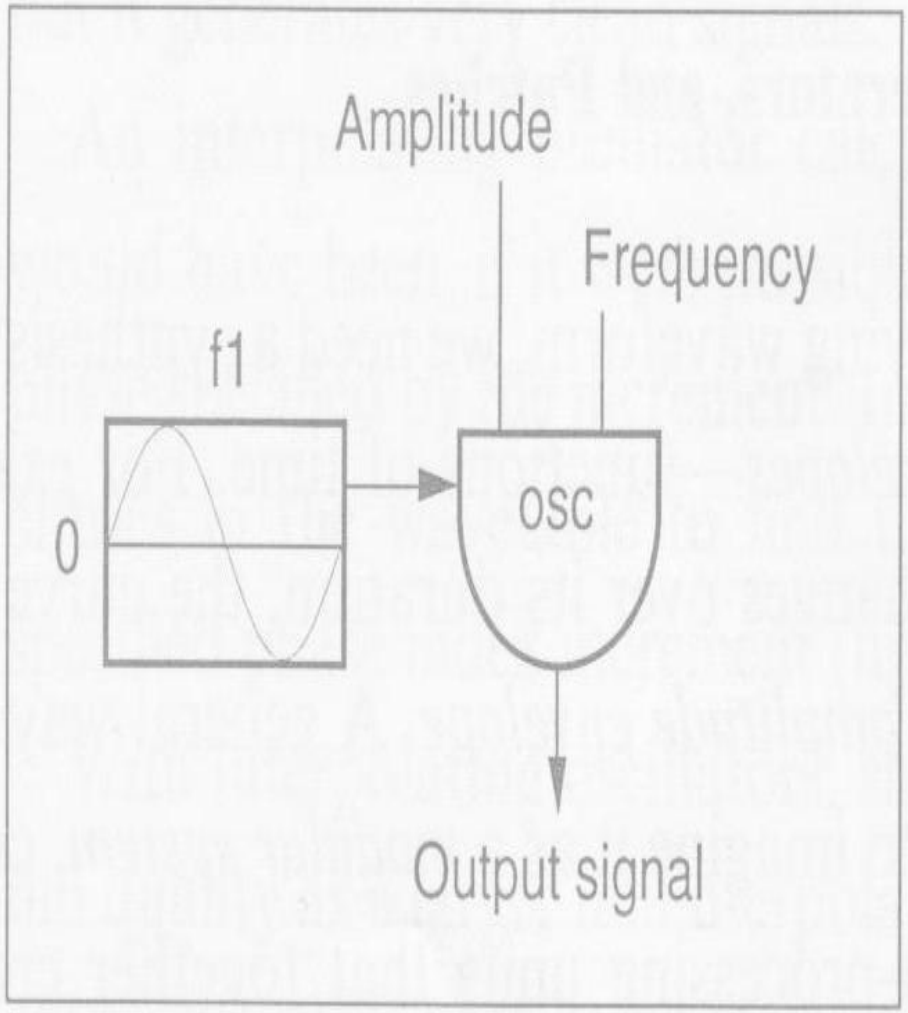


Figure 3.5 Graphical notation for an oscillator. See the text for explanation.

TABLE 3-1 Comparison of values and error produced by truncation, rounding, and linear interpolation methods of table lookup for evaluating sin θ. Phase angle θ is given in degrees, and the table length is assumed to be 360. θ is incremented by 0.75° for each sample. The truncation method produces sin ⌊θ⌋, the rounding method produces sin ⌊θ + 0.5⌋, and the linear interpolation method produces sin ⌊θ⌋ + (θ - ⌊θ⌋) [sin (⌊θ⌋ + 1) - sin ⌊θ⌋], where ⌊x⌋ stands for "floor" of x (the largest integer not larger than x). All values shown are rounded to four decimal places. * indicates an extremely small nonzero value.

Table Lookup Method Error Comparison

Sample	Phase Angle	Correct Value	Method			Error		
			Truncation	Rounding	Interpolation	Truncation	Rounding	Interpolation
0	0.00	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1	0.75	0.0131	0.0000	0.0175	0.0131	–0.0131	0.0044	*
2	1.50	0.0262	0.0175	0.0349	0.0262	–0.0087	0.0087	*
3	2.25	0.0393	0.0349	0.0349	0.0393	–0.0044	–0.0044	*
4	3.00	0.0523	0.0523	0.0523	0.0523	0.0000	0.0000	0.0000

TABLE 3-2 Approximate signal-to-error noise ratios for truncation, rounding and interpolation table lookup methods as a function of wavetable length. For sound synthesis work, the figures for the rounding method ($SN_e R_{rnd}$) generally apply to both the rounding and truncation methods. All of these figures assume that wave table entries are specified to a precision sufficient to avoid quantization errors.

Wave Table Length	$SN_e R_{trunc}$ (dB)	$SN_e R_{rnd}$ (dB)	$SN_e R_{interp}$ (dB)
32	19	25	48
64	25	31	60
128	31	37	72
256	37	43	84
512	43	49	96
1,024	49	55	108
2,048	55	61	120
4,096	61	67	132
8,192	67	73	144
16,384	73	79	156
32,768	79	85	168
65,536	85	91	180

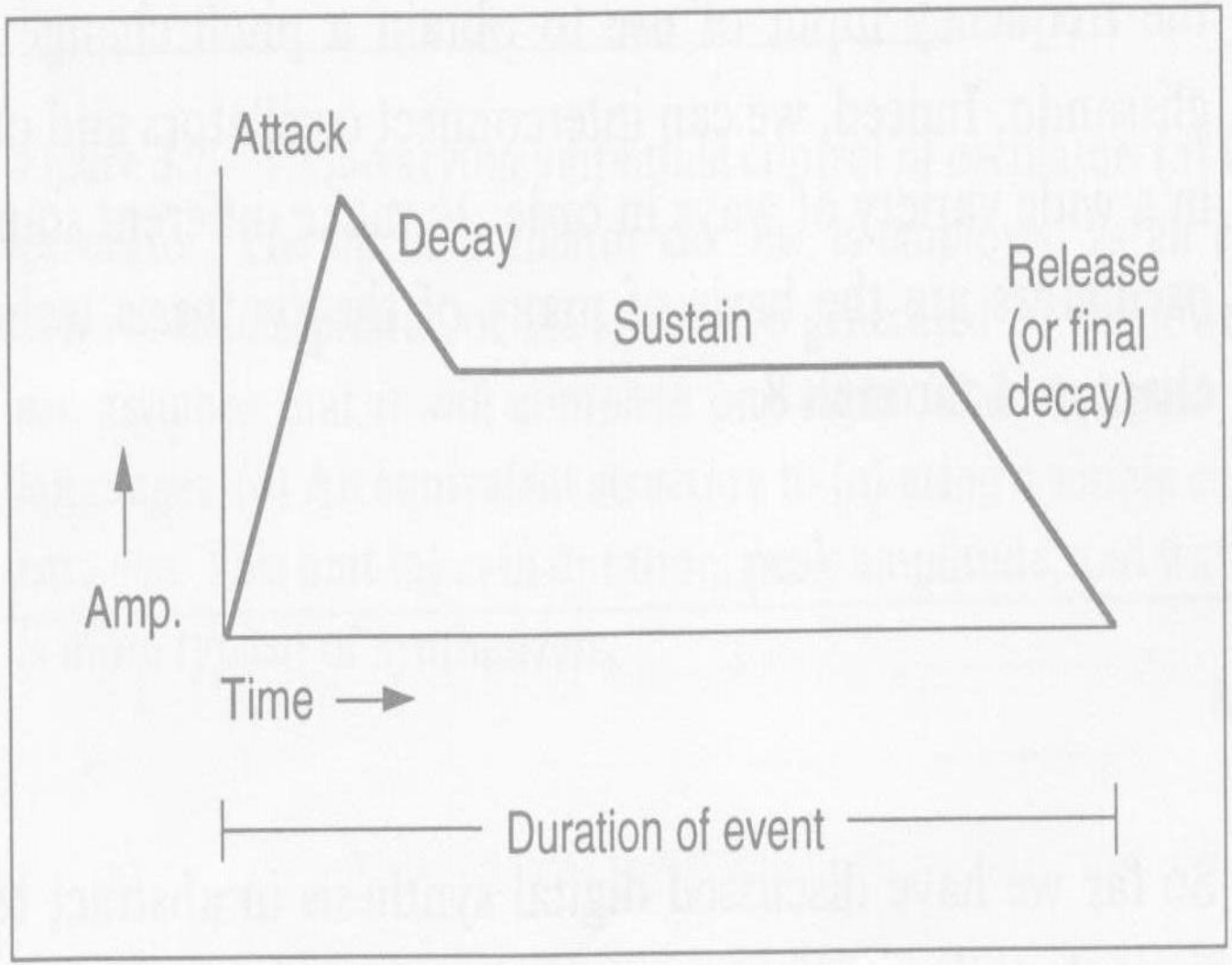


Figure 3.6 Graph of a simple ADSR amplitude envelope, showing the way the amplitude of a note changes over its duration.

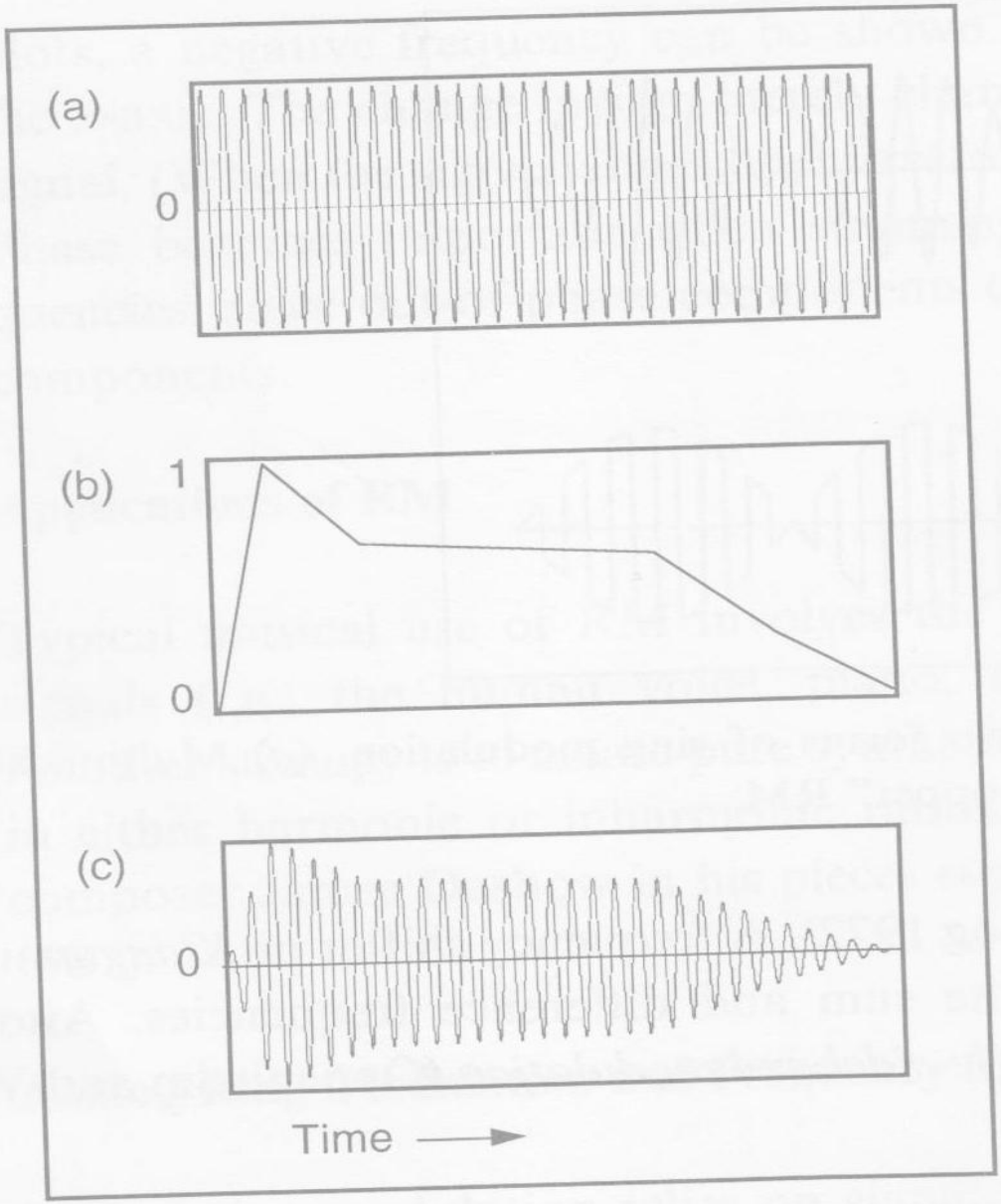


Figure 6.5 Applying an envelope to a signal is a simple case of infra-audio AM. The sine wave signal in (*a*) is multiplied by the envelope signal in (*b*) to produce the enveloped signal in (*c*).

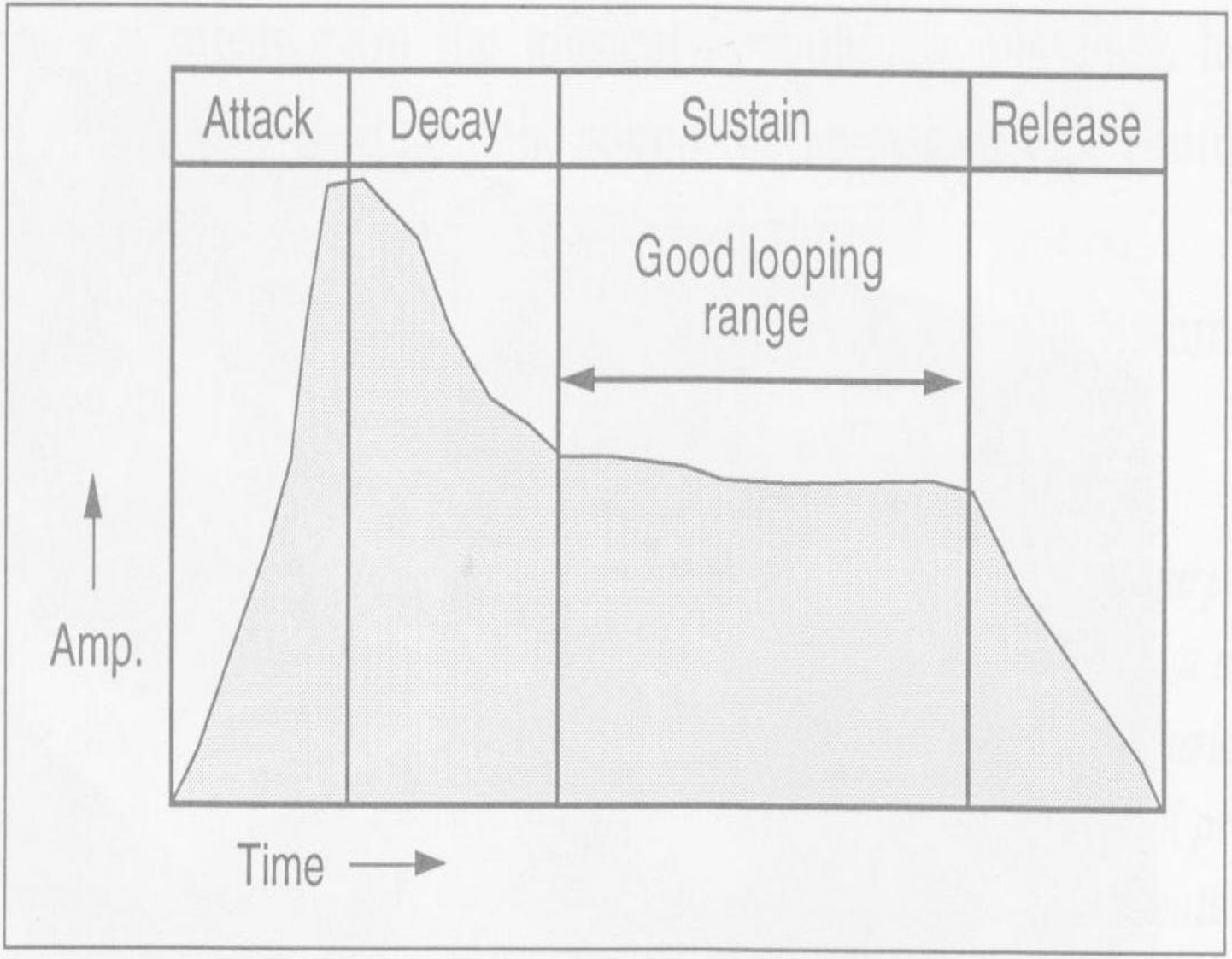


Figure 4.4 Sound with a characteristic ADSR amplitude envelope. The best area for a smooth loop is the sustained portion.

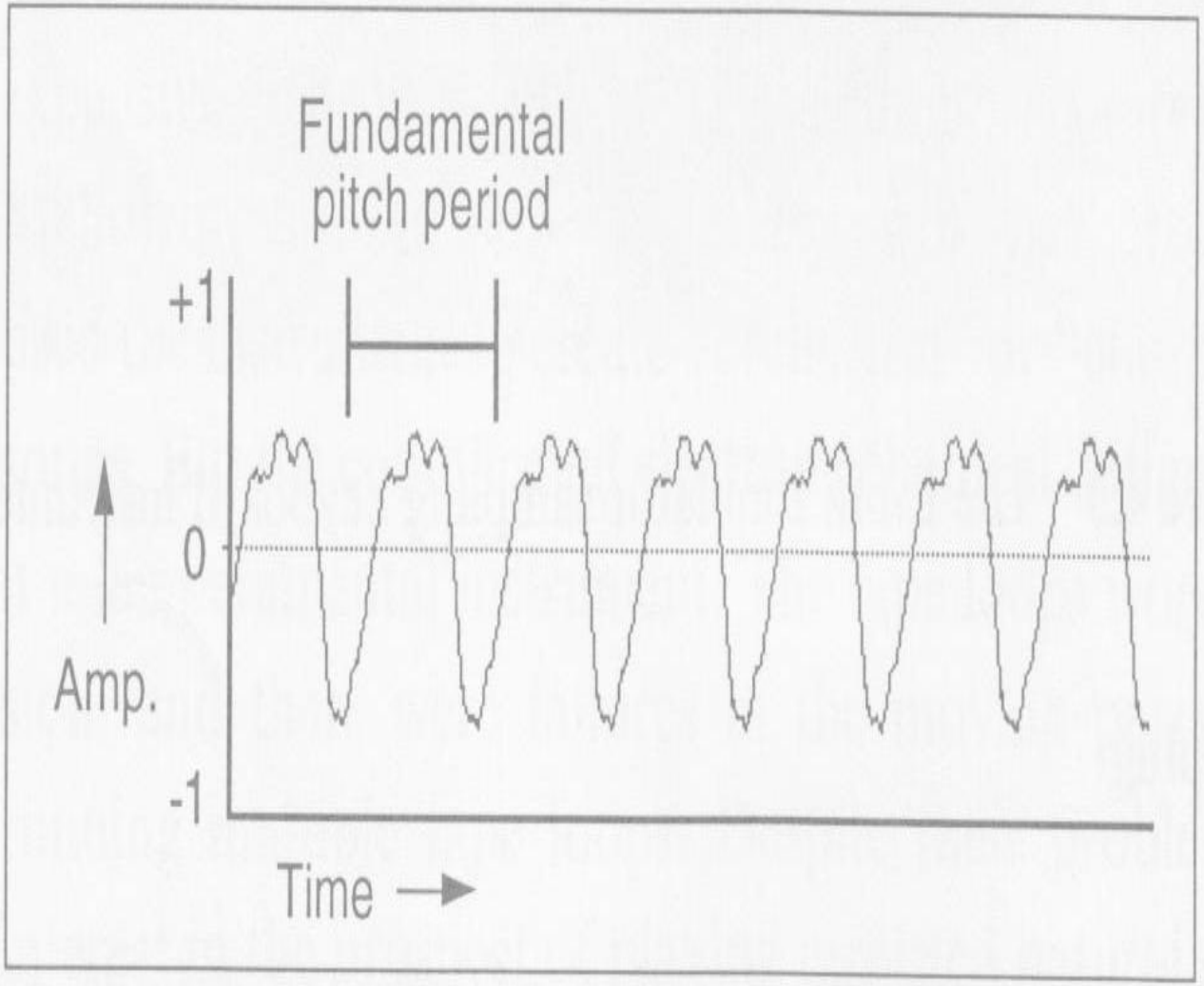


Figure 4.5 The fundamental pitch period is equal to one cycle of a periodic waveform, in this case, a waveform emitted by an alto saxophone.

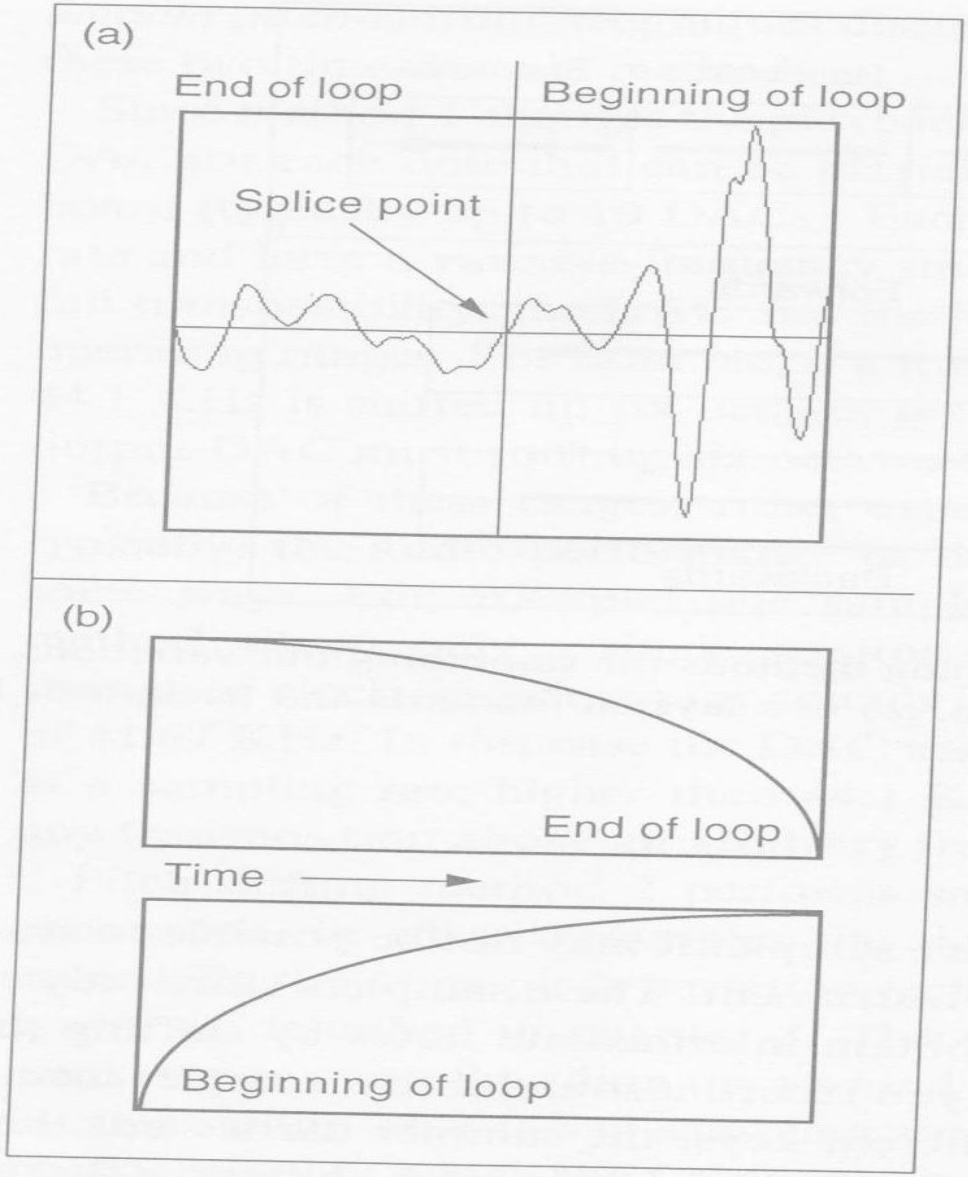


Figure 4.6 Splicing versus crossfading loops. (*a*) A vertical splice of two parts of a waveform at a common zero point. The ending point of the loop splices to the beginning of the same wavetable loop. (*b*) Crossfade looping can be viewed as a fade out of the end of the loop overlapped by a fade in of the beginning of the loop.

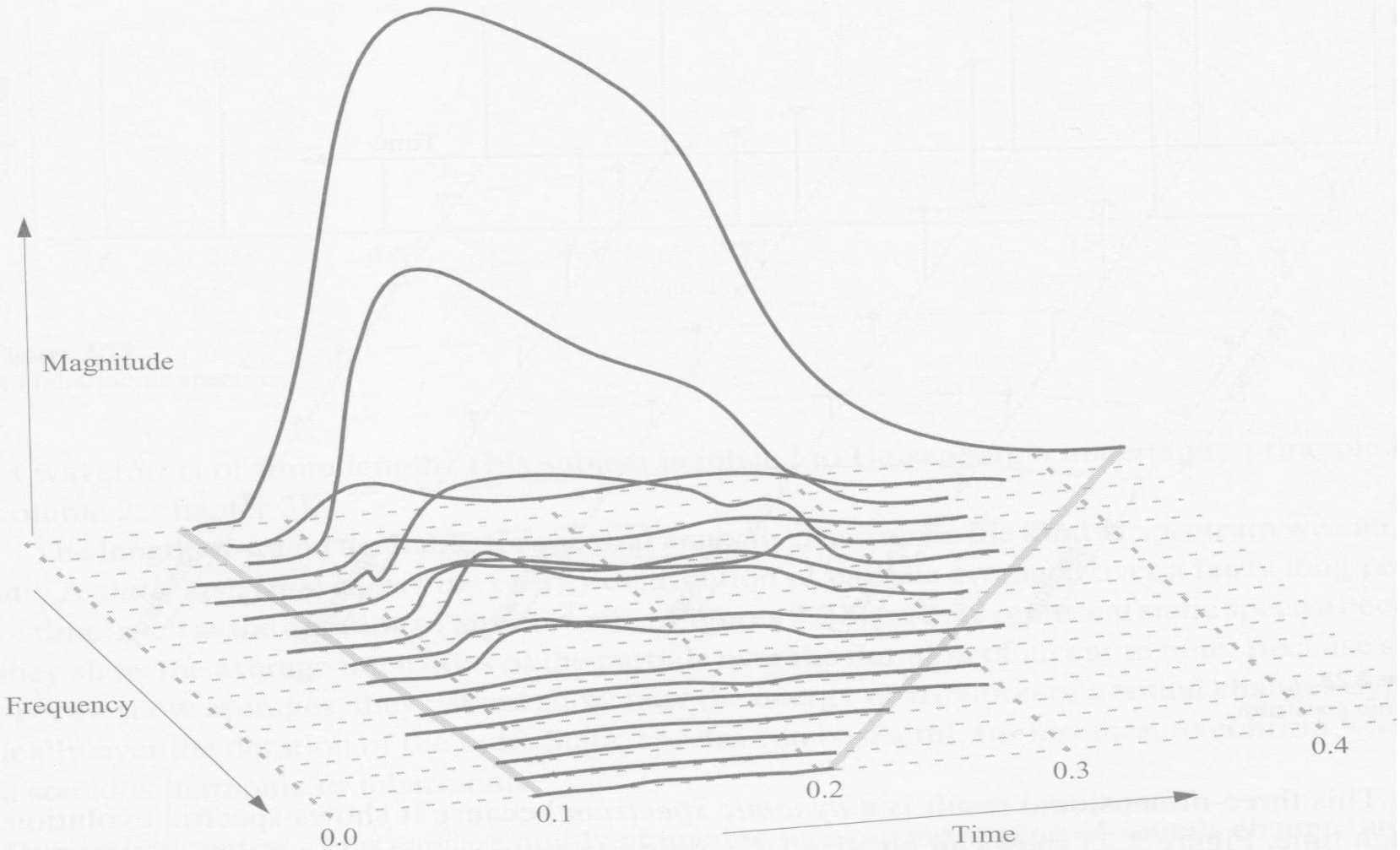


Figure 2.24
Amplitude, frequency, and time plot of a stringed instrument tone. (Adapted from a drawing in Grey 1975.)

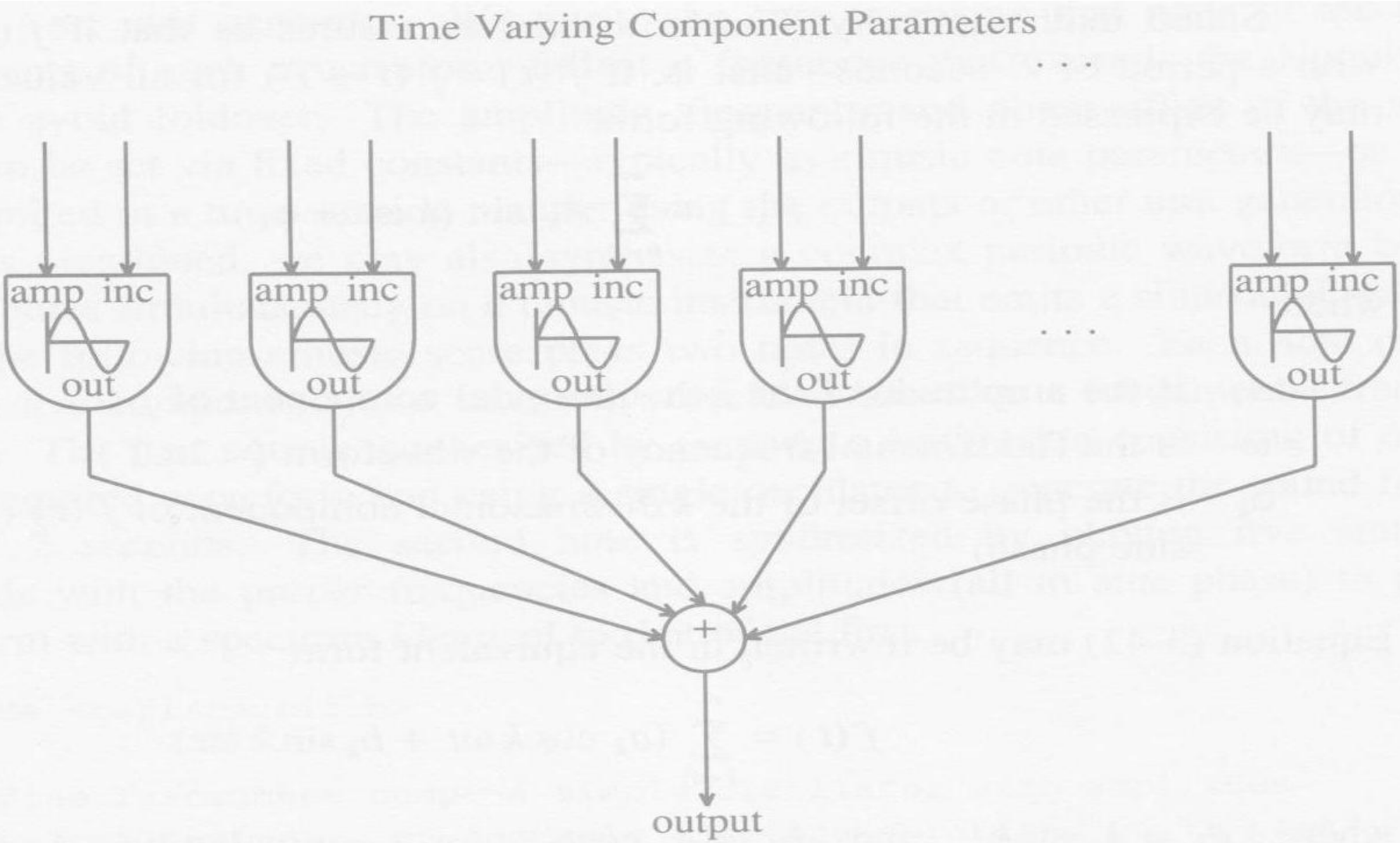


Figure 3-18 Processing model for additive synthesis. The complex sound is built up by adding together a number (possibly a large number) of sinusoidal components, each with independently varying amplitude, frequency, and phase, though in many cases the relative phase of the individual components is unspecified.

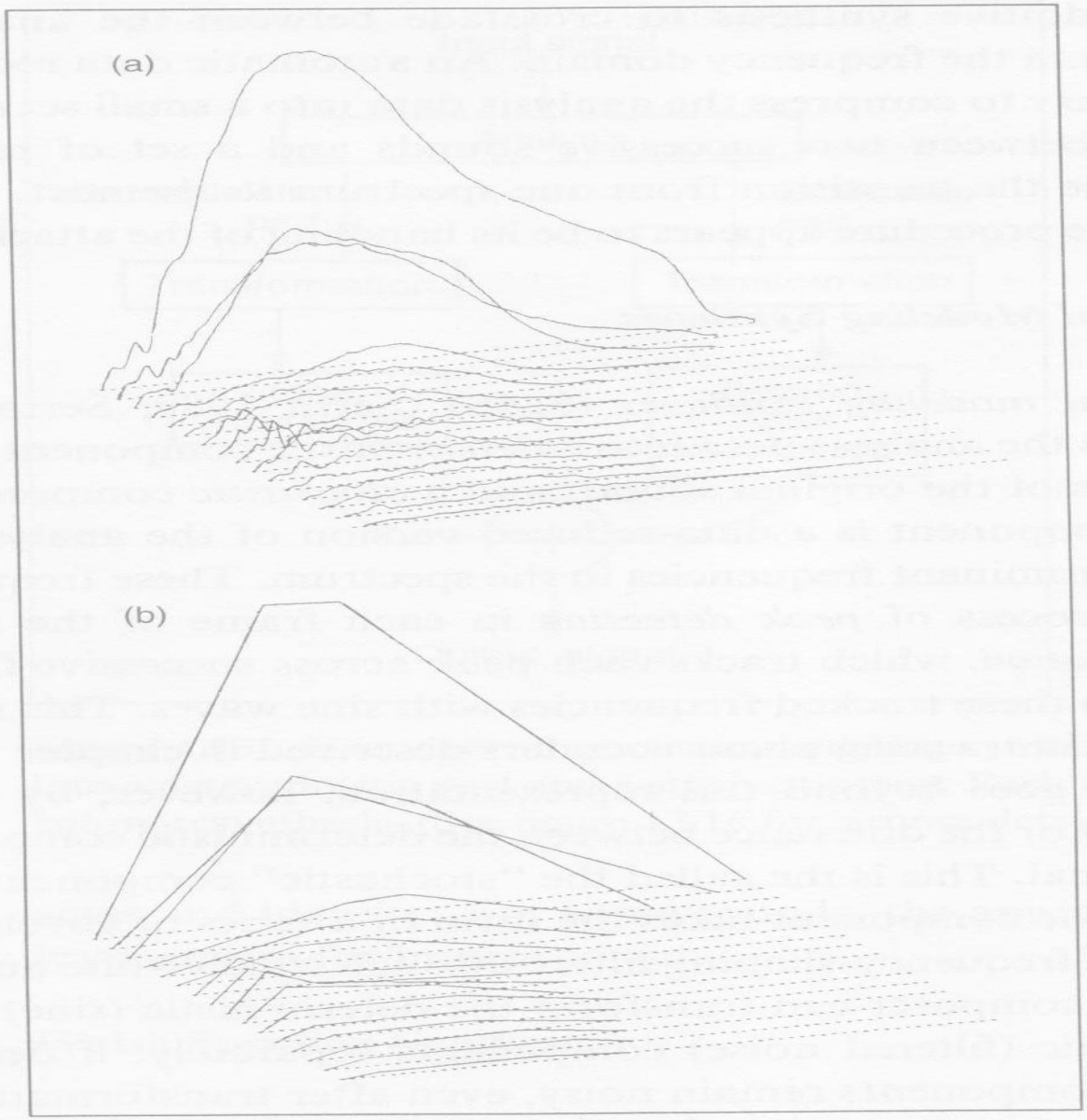
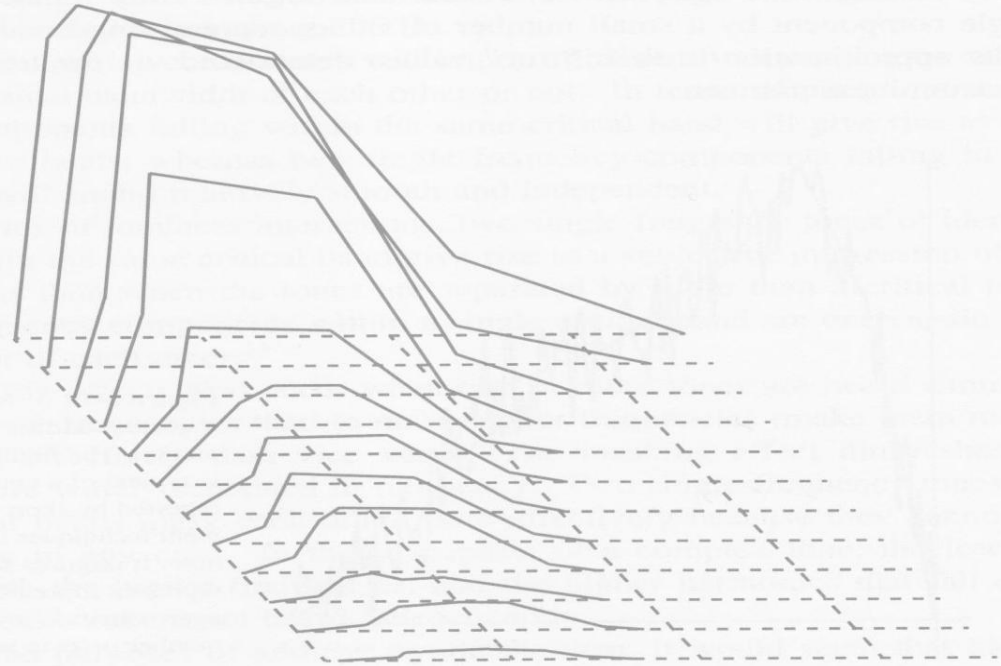


Figure 4.22 Drastic data reduction of analysis data for additive synthesis. Amplitude is plotted vertically, frequency goes from back to front, and time goes left to right. (*a*) Original frequency-vrs-time-vrs-amplitude curve of a violin tone. (*b*) The same violin tone as in (*a*), approximated with only three line segments per partial.



AMP 1 = (1, 0), (20, 305), (36, 338), (141, 288), (237, 80), (360, 0)
FREQ 1 = (1, 321), (16, 324), (32, 312), (109, 310), (317, 314), (360, 310)
AMP 2 = (3, 0), (25, 317), (39, 361), (123, 295), (222, 40), (326, 0)
FREQ 2 = (2, 0), (3, 607), (16, 657), (24, 621), (133, 621), (275, 628), (326, 628), (327, 0)
AMP 3 = (2, 0), (19, 100), (34, 369), (111, 342), (207, 41), (273, 0)
FREQ 3 = (2, 977), (5, 782), (15, 987), (24, 932), (128, 932), (217, 936), (273, 945), (275, 0)
AMP 4 = (1, 0), (3, 0), (24, 113), (29, 257), (118, 231), (187, 35), (235, 0)
FREQ 4 = (1, 0), (2, 0), (3, 718), (16, 1335), (24, 1243), (108, 1240), (199, 1248), (235, 1248), (236, 0)
AMP 5 = (1, 0), (27, 52), (34, 130), (110, 126), (191, 13), (234, 0)
FREQ 5 = (1, 1225), (9, 1569), (12, 1269), (21, 1573), (37, 1553), (97, 1552), (181, 1556), (234, 1566), (235, 0)
AMP 6 = (1, 0), (46, 83), (64, 100), (100, 100), (189, 11), (221, 0)
FREQ 6 = (1, 1483), (12, 1572), (23, 1988), (33, 1864), (114, 1864), (177, 1868), (221, 1879), (222, 0)
AMP 7 = (1, 0), (37, 39), (45, 77), (110, 79), (176, 11), (205, 0), (207, 0)
FREQ 7 = (1, 1792), (9, 1612), (29, 2242), (36, 2174), (93, 2176), (126, 2170), (205, 2188), (207, 0)
AMP 8 = (2, 0), (28, 17), (43, 71), (109, 66), (172, 8), (201, 0)
FREQ 8 = (2, 1590), (29, 2539), (36, 2491), (114, 2481), (153, 2489), (201, 2491), (203, 0)
AMP 9 = (2, 0), (29, 16), (43, 53), (54, 66), (105, 64), (165, 7), (191, 0)
FREQ 9 = (2, 1993), (25, 2121), (32, 2821), (37, 2796), (84, 2798), (105, 2792), (191, 2797), (192, 0)
AMP 10 = (1, 0), (27, 6), (41, 25), (56, 29), (72, 22), (95, 24), (180, 0)
FREQ 10 = (1, 1742), (12, 1849), (32, 3131), (37, 3111), (114, 3103), (164, 3116), (180, 3116), (181, 0)
AMP 11 = (2, 0), (37, 6), (55, 25), (88, 29), (114, 28), (164, 3), (186, 0)
FREQ 11 = (2, 1398), (41, 3419), (42, 3419), (91, 3419), (106, 3406), (150, 3421), (186, 3421), (187, 0)
AMP 12 = (7, 0), (39, 3), (43, 8), (88, 11), (118, 9), (138, 3), (165, 0)
FREQ 12 = (6, 0), (7, 1806), (23, 2942), (36, 2759), (37, 3746), (50, 3723), (84, 3731), (110, 3721), (156, 3741), (165, 3620), (167, 0)

Figure 3-20 Perspective plot and numeric breakpoint information for the piecewise linear approximation to a trumpet tone. Data for each of 12 amplitude and frequency curves is shown (the plot shows only the amplitude curves). The first number of each number pair is time in milliseconds. For the frequency curves, the second number is in hertz; for the amplitude curves, the second number is in arbitrary (linear) units.

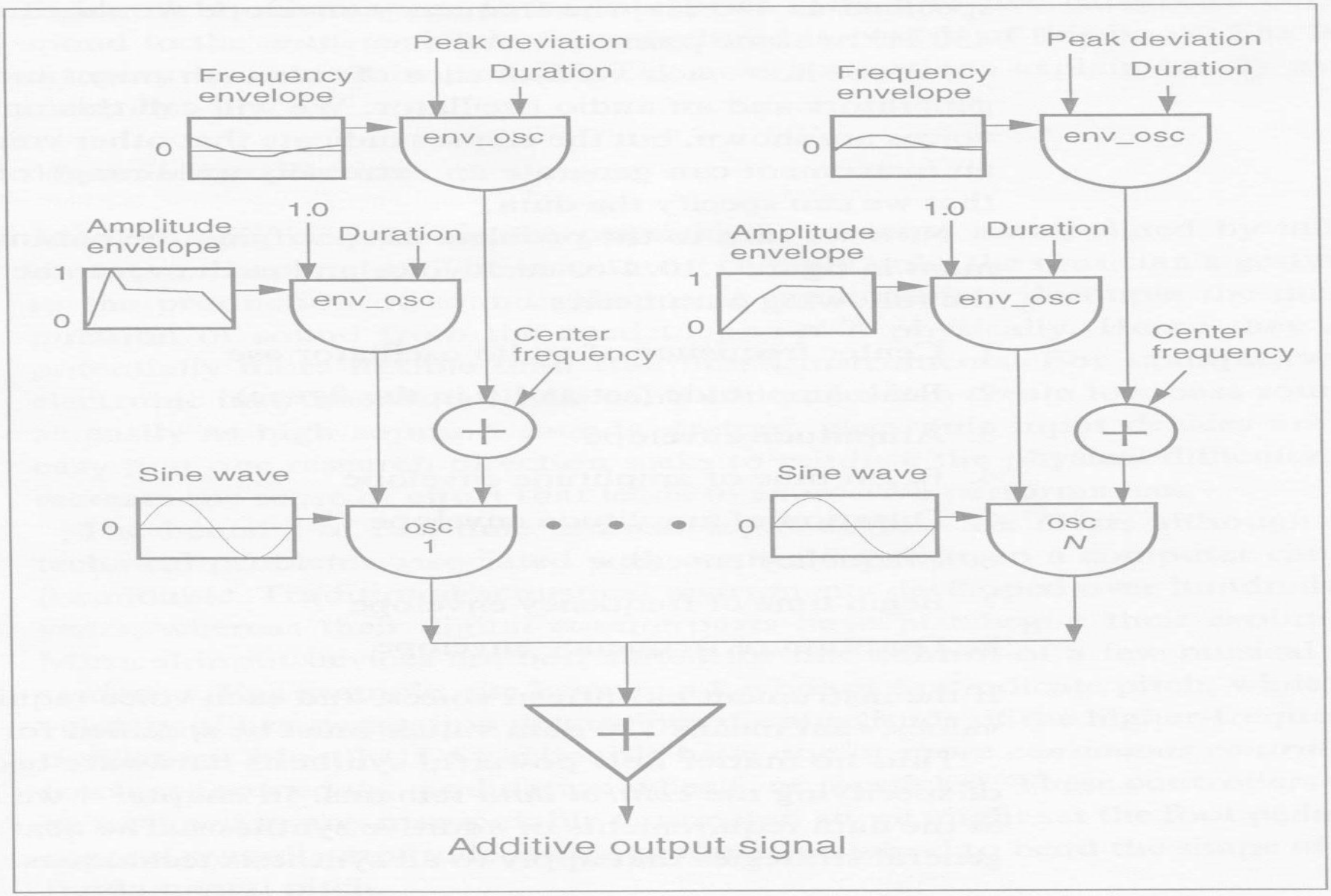


Figure 3.10 The patch shown in figure 3.7 expanded to form a simplified instrument for additive synthesis. Each sine oscillator is modified by an amplitude and frequency envelope. The outputs of many sine oscillators are added together to make one sample. More three-oscillator units might be added to this patch to make more complicated sounds.

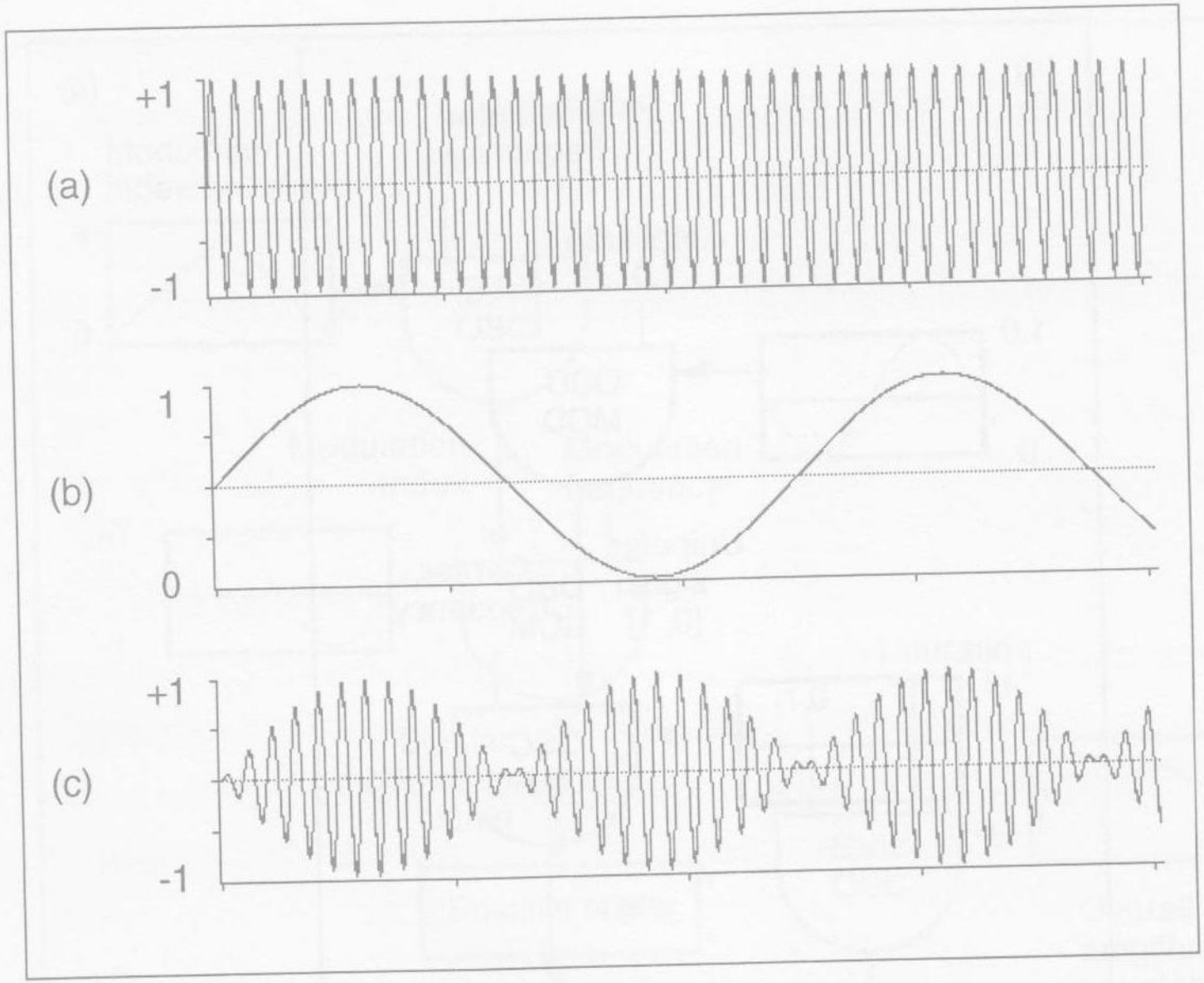


Figure 6.7 Time-domain view of audio frequency AM. The 1 KHz sine wave signal in (*a*) is modulated by the 40 Hz sine wave signal in (*b*) to produced the amplitude modulated signal in (*c*).

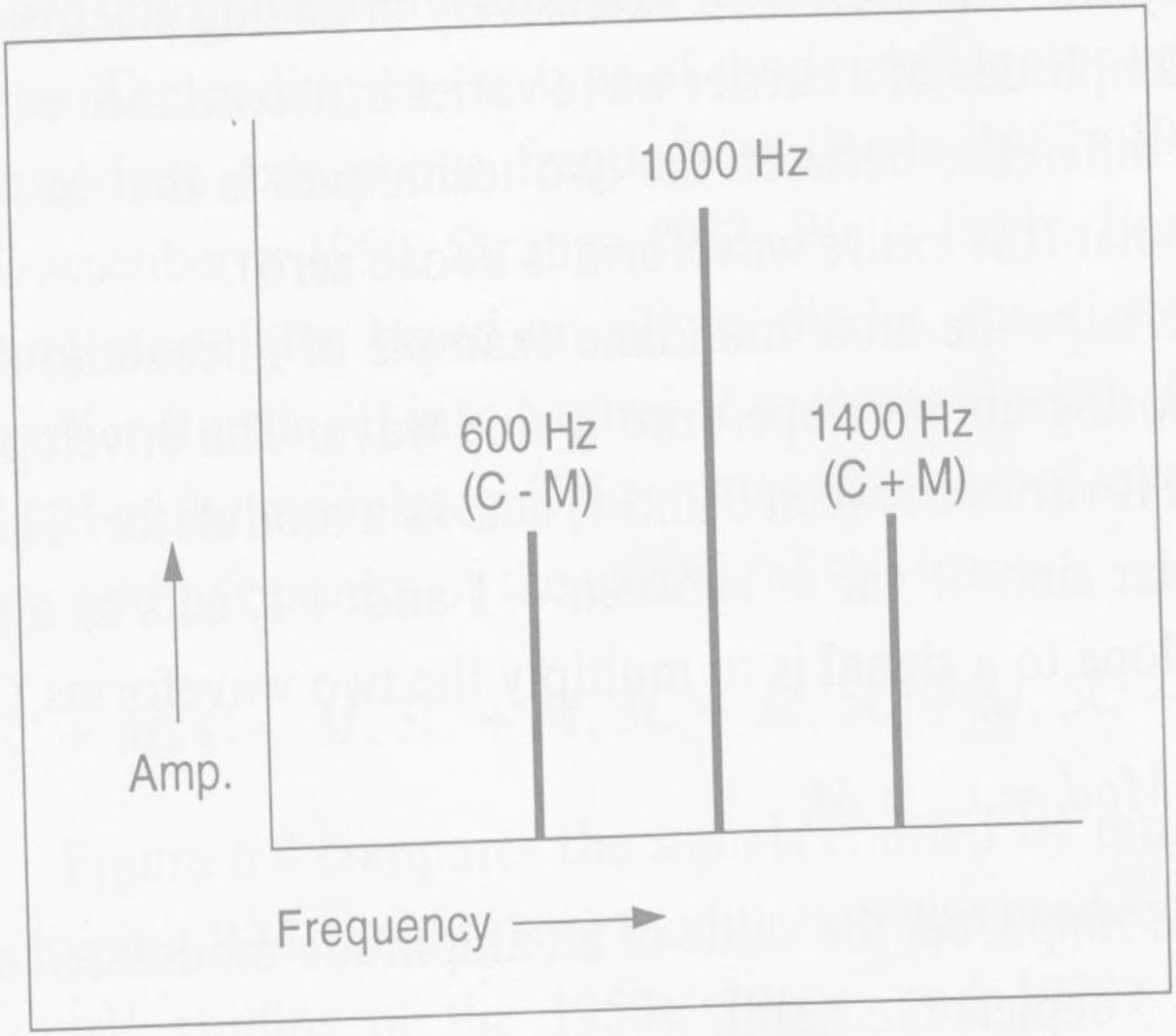
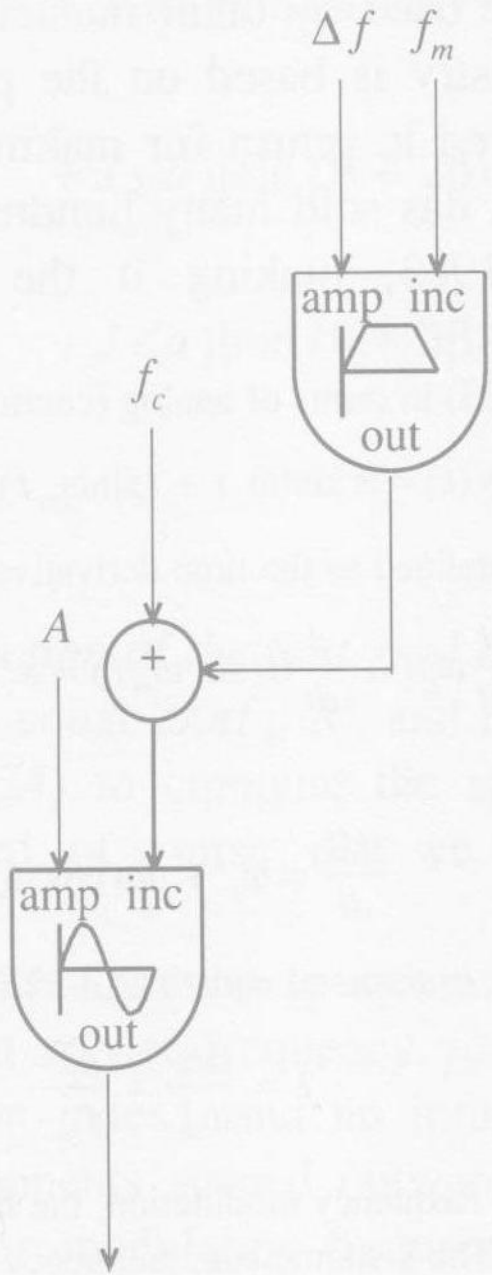
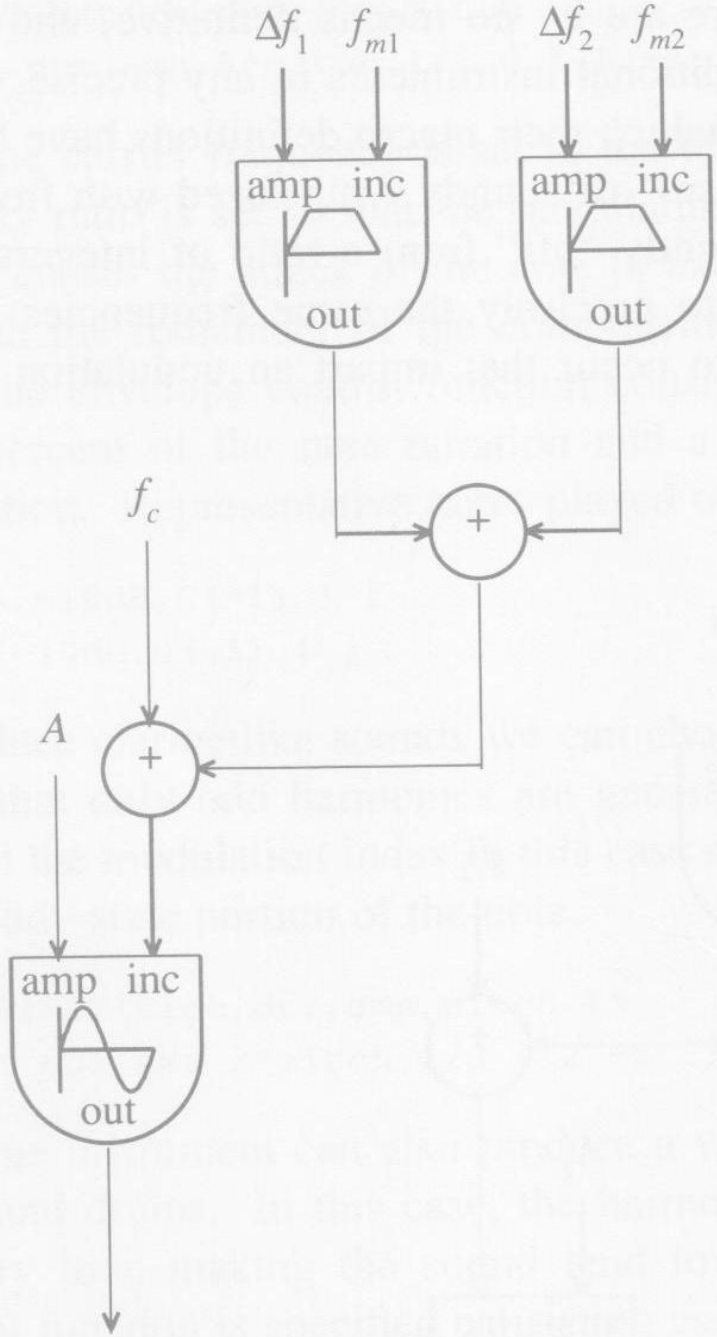


Figure 6.6 Spectrum produced by AM of a 1KHz sine wave by a 400 Hz sine wave. The two sidebands are at sum and difference frequencies around the carrier frequency. The amplitude of the each of the sidebands is *index*/2.



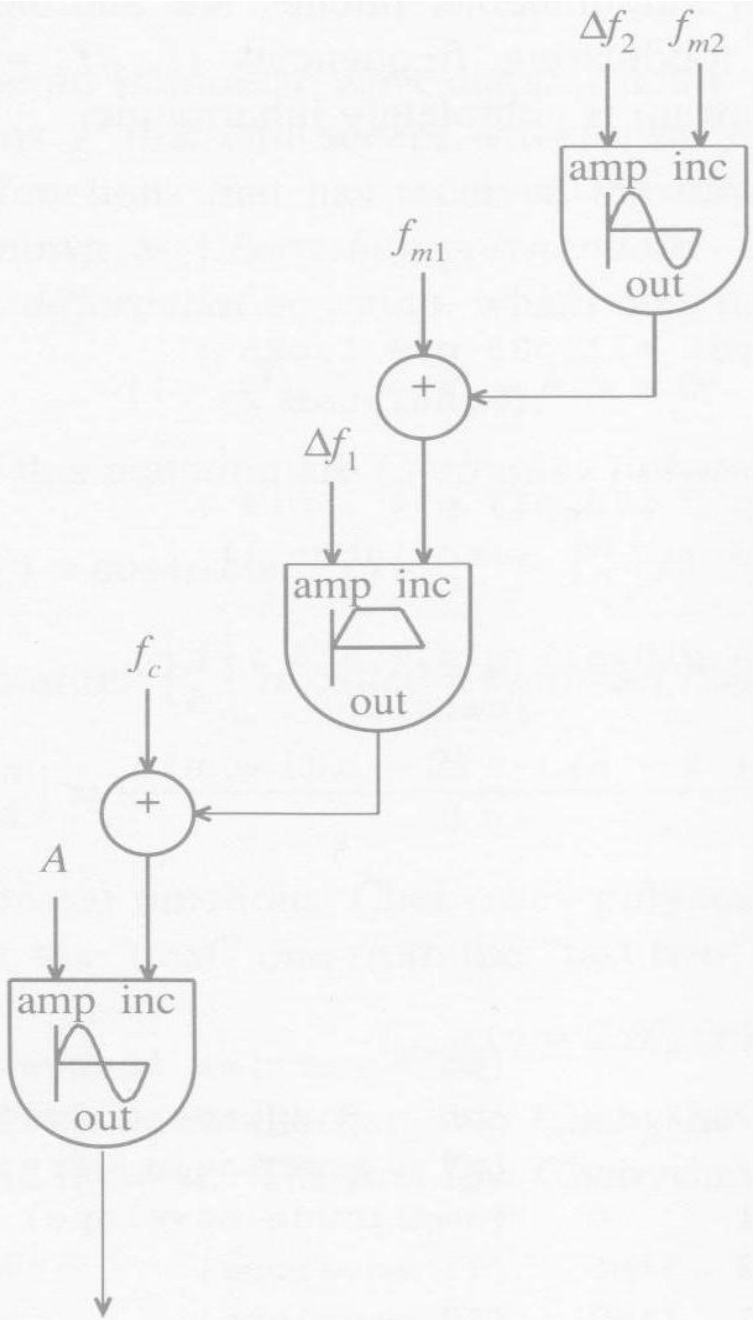
$$y(n) = A\sin(2\pi n f_c/R + I\sin 2\pi n f_m/R)$$

Figure 3-45 Definition of the basic frequency modulation process. Output signal $y(n)$ is determined by four parameters, any of which may be time-varying: A is the peak amplitude, f_c is the carrier frequency, f_m is the modulating frequency, and $I = \Delta f/f_m$ is the modulation index. Δf is the frequency deviation, R is the sampling rate, and n is the sample index. In this figure, the sampling rate and table length are assumed to be equal so that oscillator increments are numerically equal to frequencies in hertz.



$$y(t) = A\ \sin[\omega_c t + I_1 \sin(\omega_{m1}\ t) + I_2 \sin(\omega_{m2} t)]$$

Figure 3-51 Example of double modulator frequency modulation. A single-carrier oscillator is modulated by the sum of two sinusoidal modulating oscillators, effectively causing the modulating signal to have a complex waveshape. This technique may be extended to any number of modulating oscillators, forming an output waveform in which all possible sum and difference frequencies exist.



$$y(t) = A\sin\,[\omega_c t + I_1\,\sin\omega_{m1}t + I_2\sin\omega_{m2}t)]$$

Figure 3-52 Example of cascade frequency modulation. A carrier oscillator is modulated by a frequency modulated waveform; the modulating signal therefore has an infinite number of components.

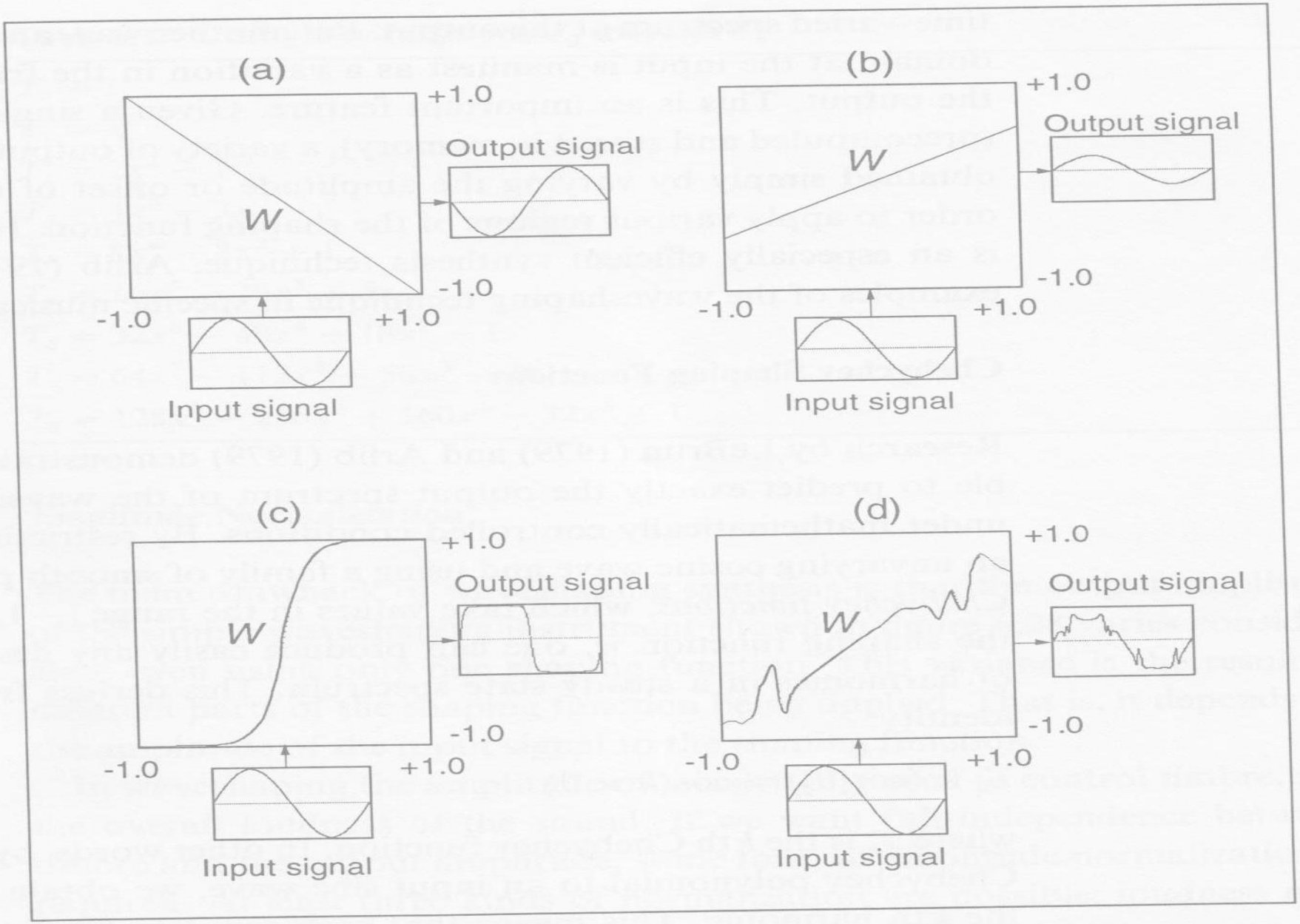


Figure 6.28 Four shaping functions. (*a*) Inversion of the input signal. (*b*) Attenuation. (*c*) Amplification of low-level signals (expansion) and clipping of high-level signals. (*d*) Complicated amplitude-sensitive distortion.

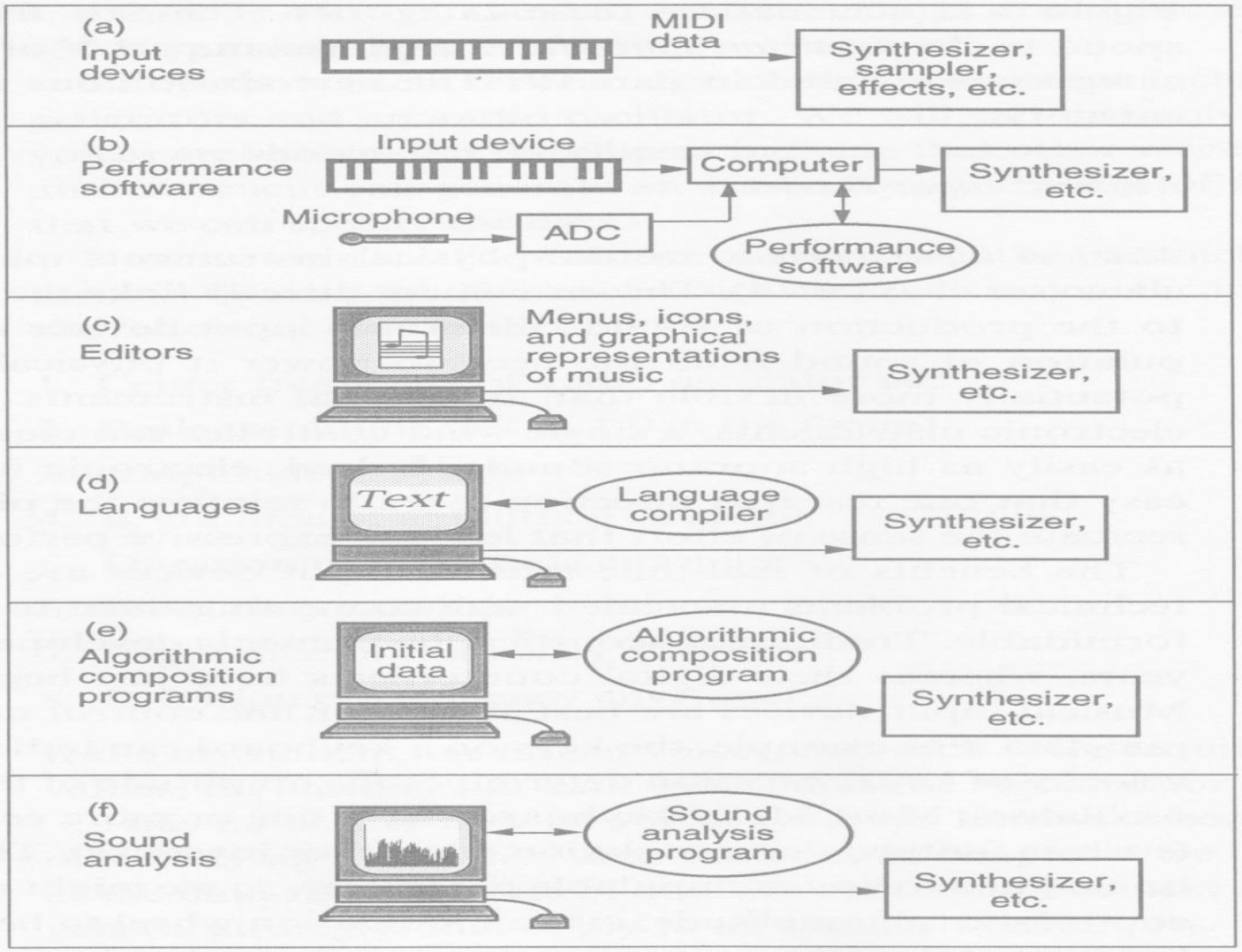
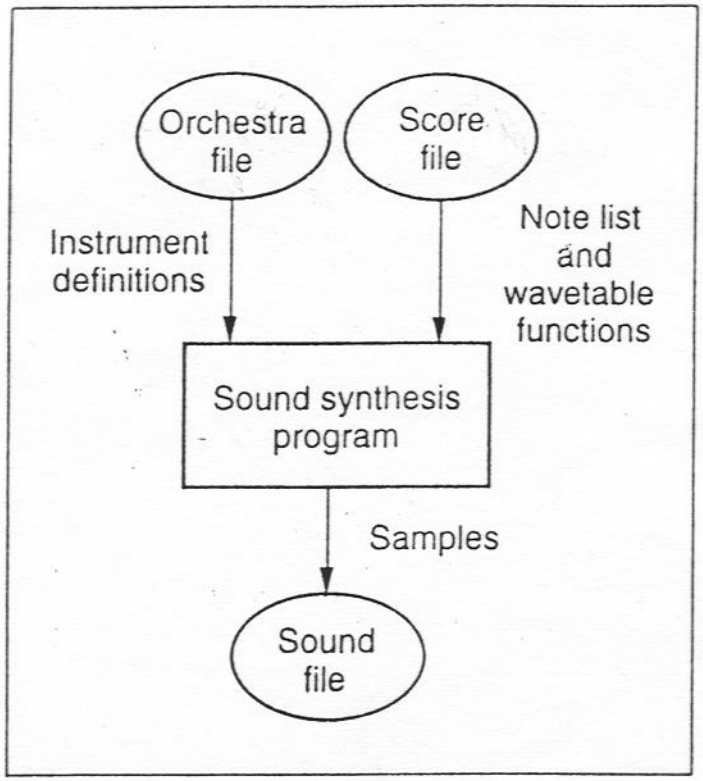
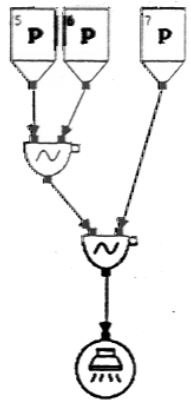


Figure 3.11 The musician's interface: six different ways of specifying synthesis data to a computer or synthesizer. (*a*) An input device can transmit the necessary data directly to a synthesizer, with or without a computer in between. (*b*) Performance software interprets the performer's gestures and may even be able to improvise. (*c*) Editors let the user build up a specification through interactive graphics techniques. (*d*) Languages encode the specification as a precise text. (*e*) Algorithmic composition programs typically require a small amount of initial parameter data from the composer before they generate music. (*f*) Sound analysis automatically derives data for modification and resynthesis from sounds fed into it.



A software synthesis program takes in score and orchestra files and generates a sound sample file.

(a)

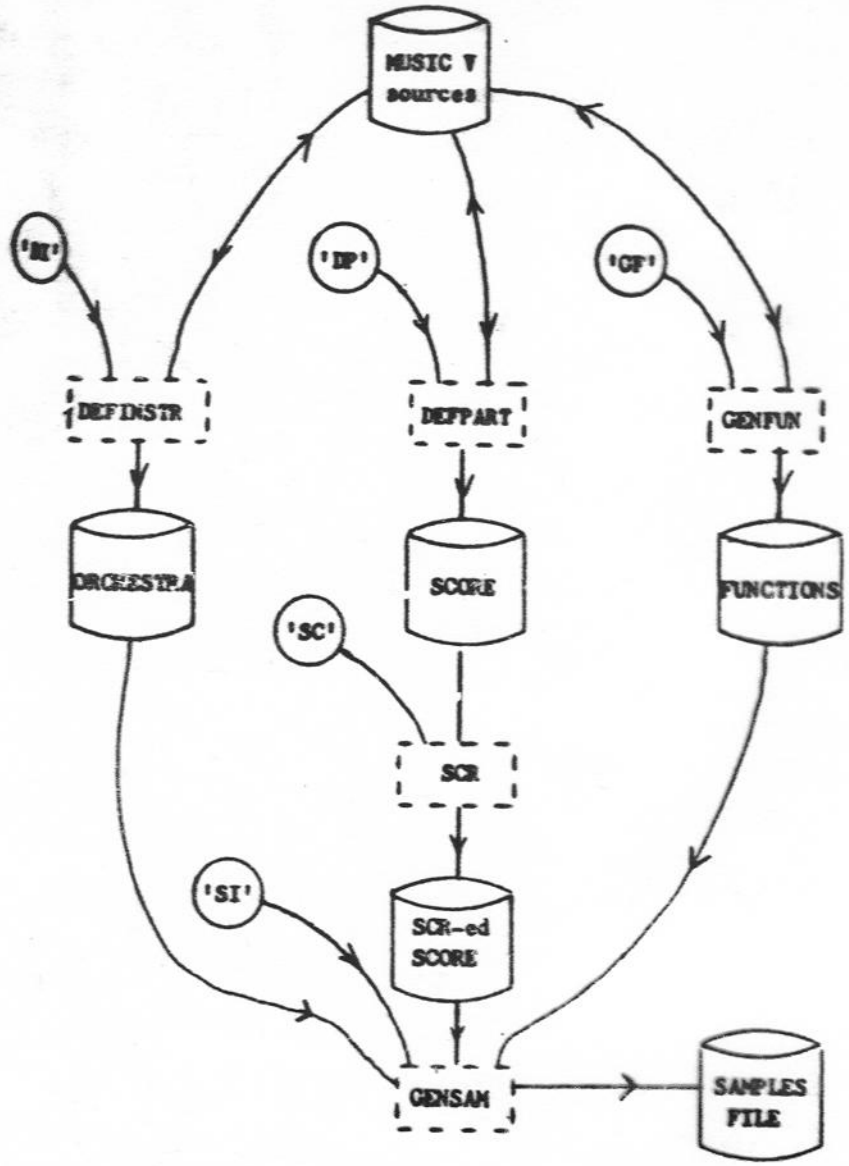


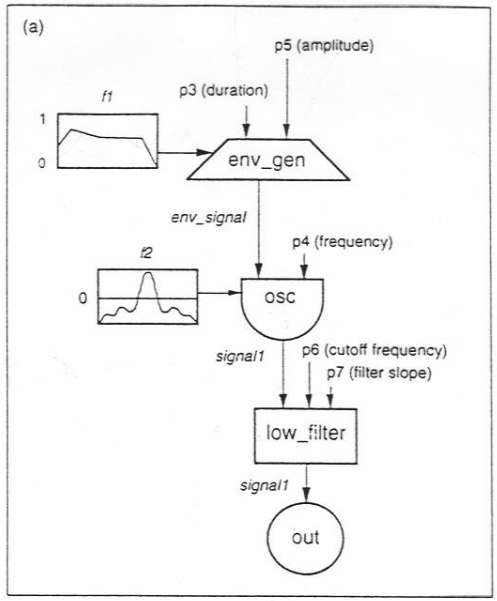
The Graphical Music V modular instrument editor. (image of a simple instrument for amplitude modulation. Each score pfield is indicated by a "P" module. The left input to the oscillator is amplitude, while the right input is frequency. The white squares in the upper right part of the oscillator are optional inputs for initial phase (unused in this example). (*b*) Code generated by compiling the instrument diagrammed in 16.32(*a*) along with Music V GEN function definitions and pfield conversion routines specified in other editors within the system. Comment statements have been typed into the automatically generated code.

(b)

```
INS  0 1 ;

 OSC  P5  P6  B2  F6  P50 ;
 OSC  B2  P7  B2  F1  P49 ;
 OUT  B2  B1 ;


COM  GEN functions for the instrument ;

COM  Modulation envelope ;
 GEN 0 1 6 256 0 1 500 64 500 128 500 193 0 256 ;

COM  Pure sine wave of 4096 samples ;
 GEN 0 2 1 4096 1000 1000 1 ;


COM  Conversion routines for the pfields ;

COM  P5 Amplitude ;
COM Conversion 200: from dB to linear amplitude ;
 PLS 0 1 1 205 ;

COM  P6 Modulation frequency ;
COM Conversion 0: from Hertz to sample increment ;
 PLS 0 1 1 6 ;

COM  Carrier frequency ;
COM Conversion 0: from Hertz to sample increment ;
 PLS 0 1 1 7 ;
```

"MUSIC V-like" PACKAGE *



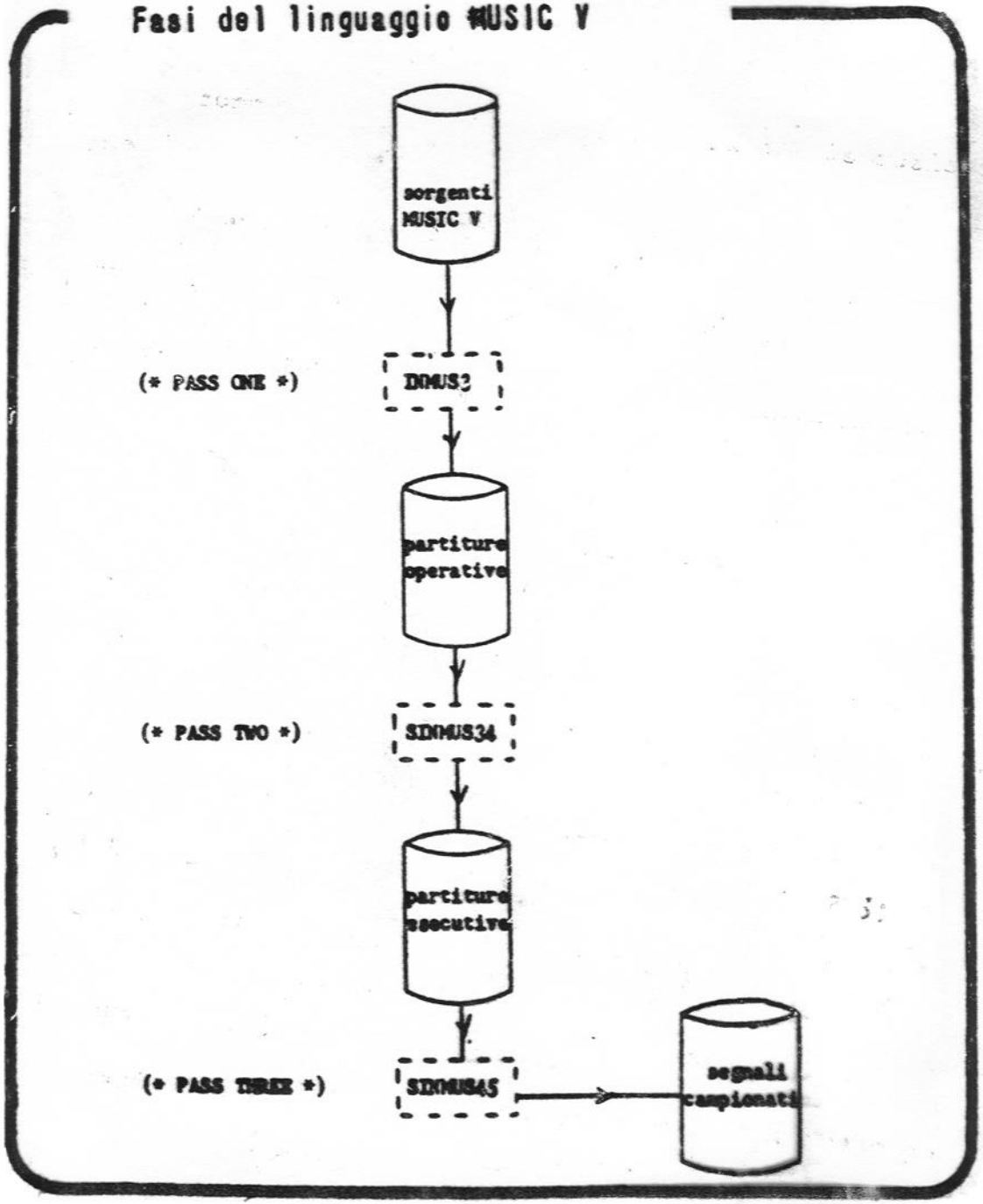


(b)

```
Instrument 1
      /* Define envelope signal */
      env_signal ← env_gen f1 p3 p5;
      /* Generate audio signal using the env_signal to control amplitude */
      signal1 ← osc  f2 p4 env_signal;
      /* Send signal1 to filter. p6 is cutoff freq., p7 is filter slope */
      signal1 ← low_filter signal1 p6 p7;
      out signal1;     /* Send signal to soundfile */
EndInstrument 1
```

Orchestra language example. Definition of an instrument with an envelope for amplitude, an oscillator, and a lowpass filter. (*a*) Graphical representation. (*b*) Textual representation. The remarks surrounded by slash-asterisk ("/*" and "*/") are comments.

Fasi del linguaggio MUSIC V



Musical Attribute	How indicated			
	orchestra: definition of individual *instruments*	*scot*: representation of conventional notation	*score*: specification of time-based event list	Audio output: combined realization of *orchestra* and *score*
Pitch	———	by octave, note names	in cycles per second (*cps*)	generated in Hertz (*cps*)
Duration	———	in beats	in beats; may be modified by *t*-statement	generated in seconds
Articulation	initialized variables (*i*-statement)	in *instrument* definition	in *instrument* definition	generated through the *orchestra* definition
Dynamics	initialized variables (*i*-statement)	by extra parameters and macros	by extra parameters and macros	generated through the *orchestra* definition
Timbre	control signals (*k*-statement)	via *instrument* definition	via *instrument* definition	generated through the *orchestra* definition

Table 7.1 *Csound*'s component parts.

Capitolo 4

Livello interpretativo

Capitolo 5

Livello simbolico

$$\begin{bmatrix} C_1 & D_1 & E & F & G & A & B & C_2 & D_2 \\ 4 & : & 5 & : & 6 & & & & \\ & & & 4 & : & 5 & : & 6 & \\ & & & & 4 & : & 5 & : & 6 \end{bmatrix}$$

Figure 3.13
Natural major scale.

Scale ratio	$\frac{1}{1}$	$\frac{9}{8}$	$\frac{5}{4}$	$\frac{4}{3}$	$\frac{3}{2}$	$\frac{5}{3}$	$\frac{15}{8}$	$\frac{2}{1}$
	C	D	E	F	G	A	B	C
Interval ratio	$\frac{9}{8}$	$\frac{10}{9}$	$\frac{16}{15}$	$\frac{9}{8}$	$\frac{10}{9}$	$\frac{9}{8}$	$\frac{16}{15}$	

Two sizes of whole steps

Figure 3.14
Natural major scale with interval sizes.

1. Find E from C:

$$\frac{E}{C} = \frac{5}{4}, \quad \text{or} \quad E = \frac{5}{4}C.$$

2. Find G from C:

$$\frac{G}{C} = \frac{6}{4}, \quad \text{or} \quad G = \frac{6}{4}C = \frac{3}{2}C.$$

3. Find F from C_2:

$$\frac{C_2}{F} = \frac{6}{4}, \quad \text{or} \quad F = \frac{4}{6}C_2 = \frac{4}{6} \cdot \frac{2}{1}C = \frac{4}{3}C.$$

4. Find A from F:

$$\frac{A}{F} = \frac{5}{4}, \quad \text{or} \quad A = \frac{5}{4}F = \frac{5}{4} \cdot \frac{4}{3} \cdot C = \frac{5}{3}C.$$

5. Find B from G:

$$\frac{B}{G} = \frac{5}{4}, \quad \text{or} \quad B = \frac{5}{4}G = \frac{5}{4} \cdot \frac{3}{2} \cdot C = \frac{15}{8}C.$$

6. Find D_2 from B:

$$\frac{D_2}{B} = \frac{6}{5}, \quad \text{or} \quad D_2 = \frac{6}{5}B = \frac{6}{5} \cdot \frac{15}{8} \cdot C = \frac{90}{40}C = \frac{9}{4}C.$$

7. Find D:

$$\frac{D_2}{D} = \frac{2}{1}, \quad \text{or} \quad D = \frac{9}{8}C.$$

$$\begin{bmatrix} C_1 & D_1 & E\flat & F & G & A\flat & B\flat & C_2 & D_2 \\ 10 & : & 12 & : & 15 & & & & \\ & & & 10 & : & 12 & : & 15 & \\ & & & & 10 & : & 12 & : & 15 \end{bmatrix}$$

Figure 3.16
Just minor scale.

Table 3.6
Comparison of Natural and Equal-Tempered Chromatic Intervals

Degree	Name	Error	Degree	Name	Error
1	Unison	0.0	7	Tritone	–9.7763
2	Minor second	–11.731	8	Perfect fifth	–1.955
3	Major second	–3.910	9	Minor sixth	–13.686
4	Minor third	–15.641	10	Major sixth	15.641
5	Major third	13.686	11	Minor seventh	3.910
6	Perfect fourth	1.955	12	Major seventh	11.730

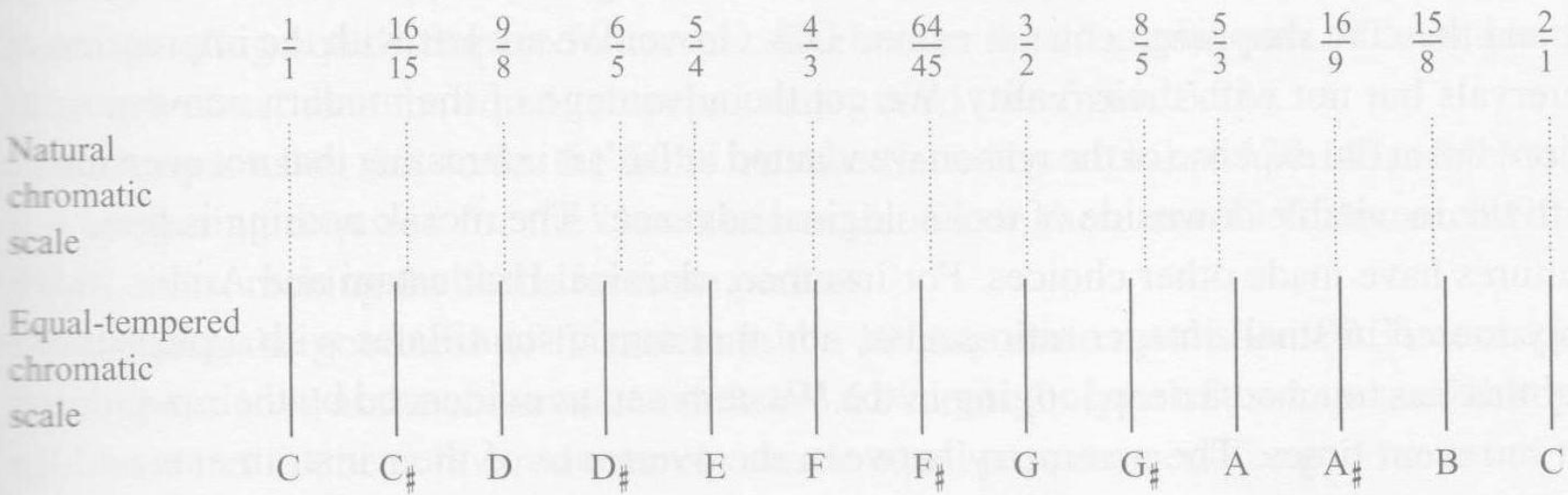


Figure 3.20
Natural and equal-tempered chromatic intervals.

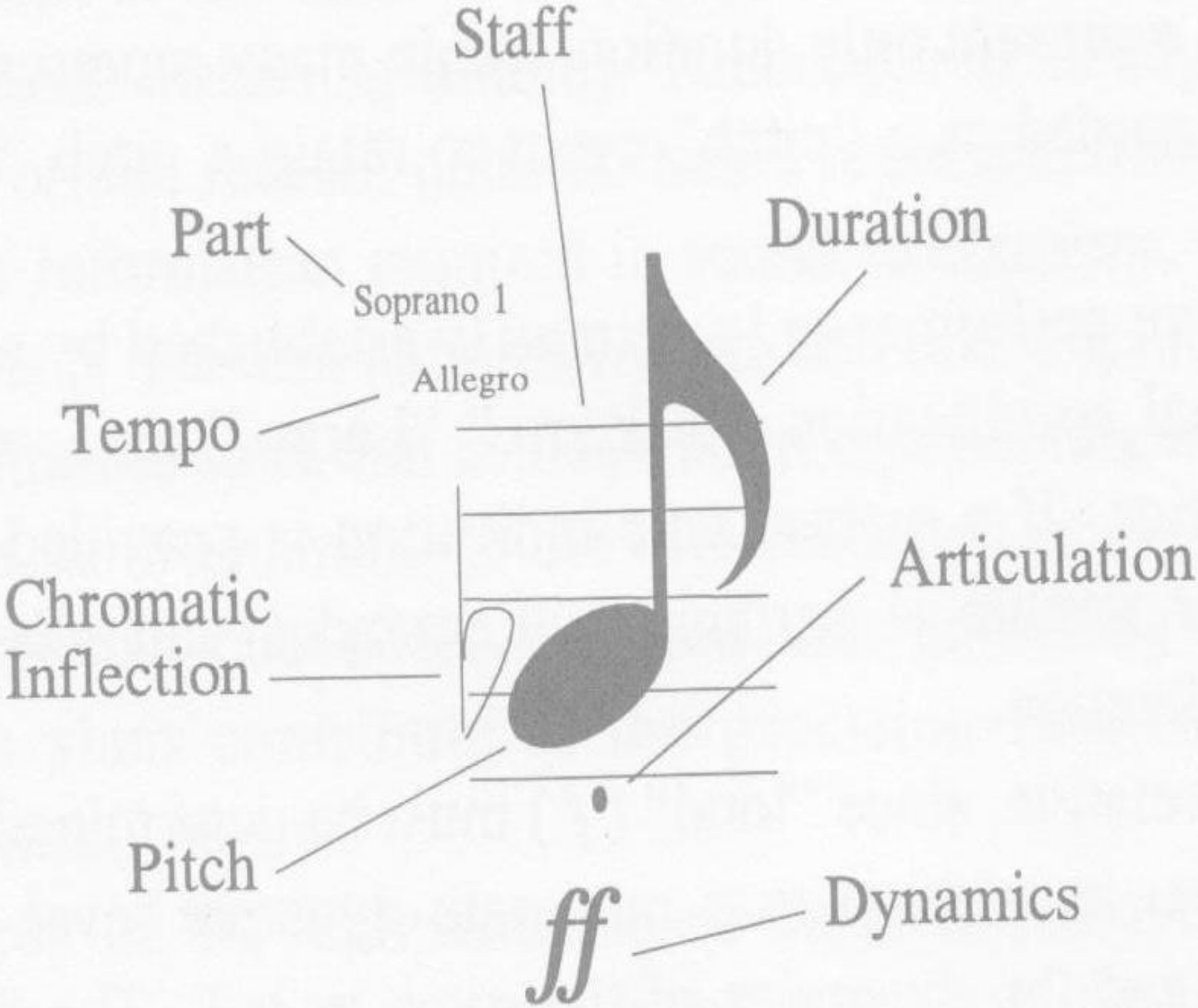
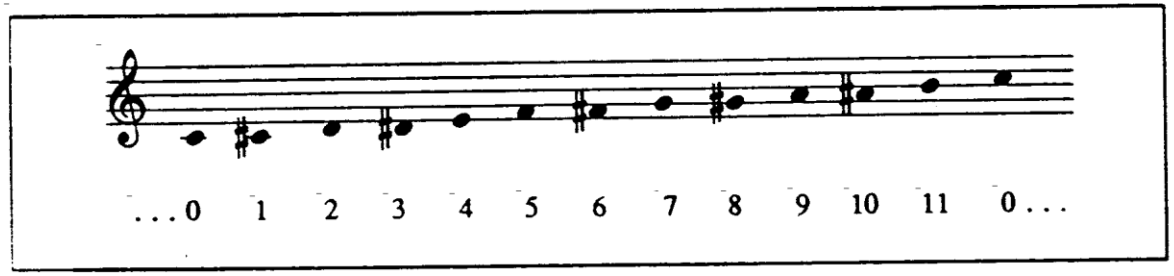


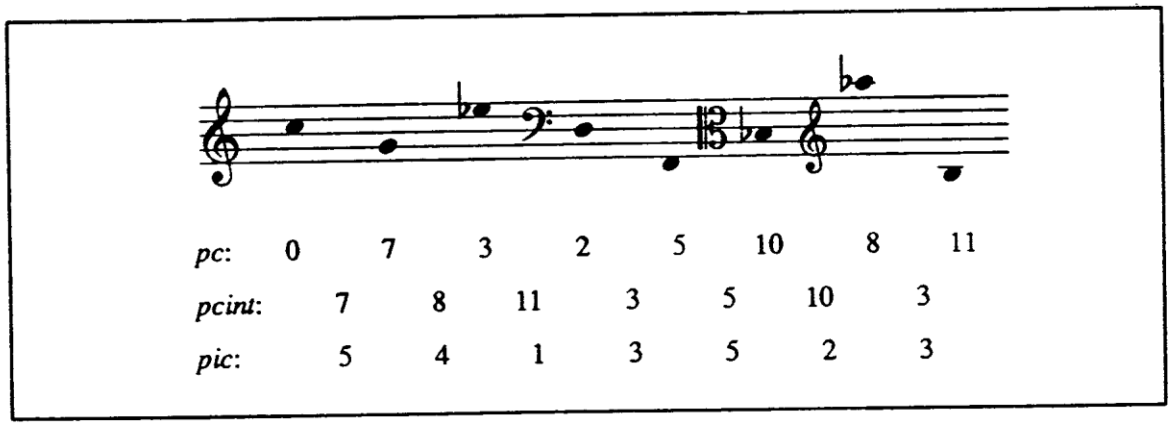
Figure 1.1 Some attributes of (or associated with) a single note.



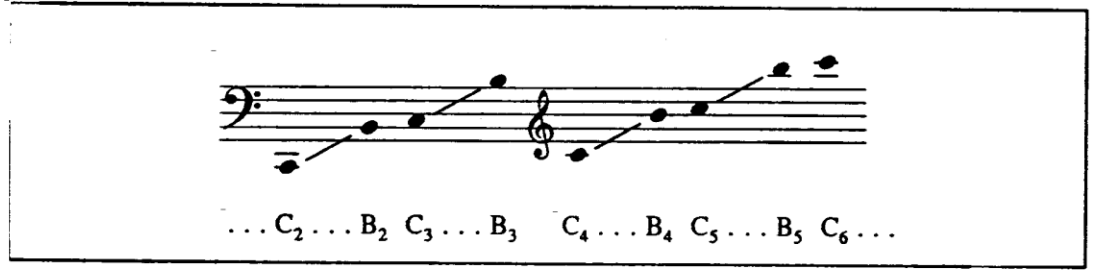
Pitch-Class Integers

Name	*pcint*	*pcint*	*Name*
perfect prime	0	0	perfect octave
minor second	1	11	major seventh
major second	2	10	minor seventh
minor third	3	9	major sixth
major third	4	8	minor sixth
perfect fourth	5	7	perfect fifth
augmented fourth	6	6	diminished fifth

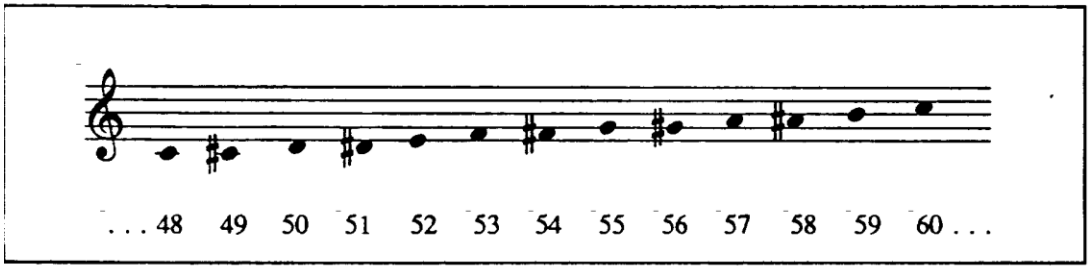
Intervals in the Pitch-Class System



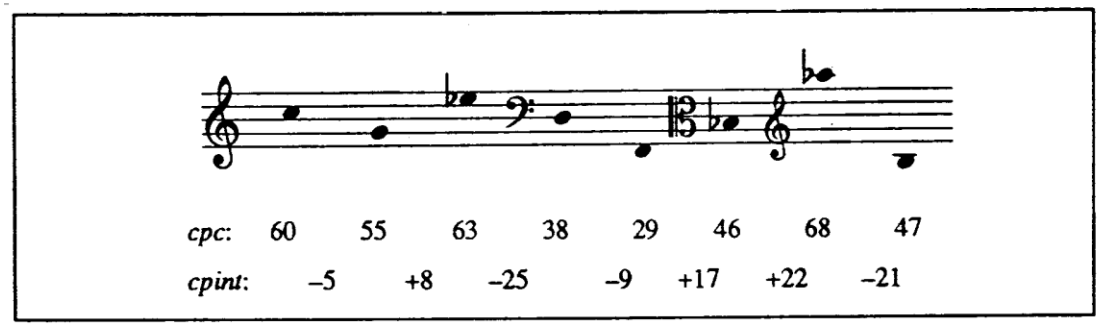
Examples of Pitch-Class Integers, Intervals, and Interval Classes



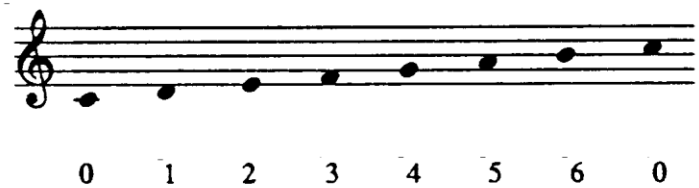
Standard Octave Numbers



The Continuous Pitch Code Scale



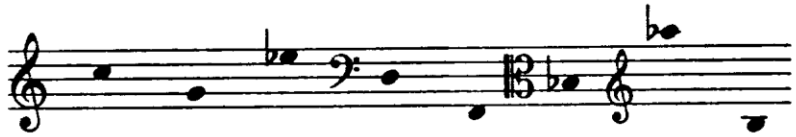
Examples of Continuous Pitch Codes and Continuous Pitch Intervals



Name-Class Integers

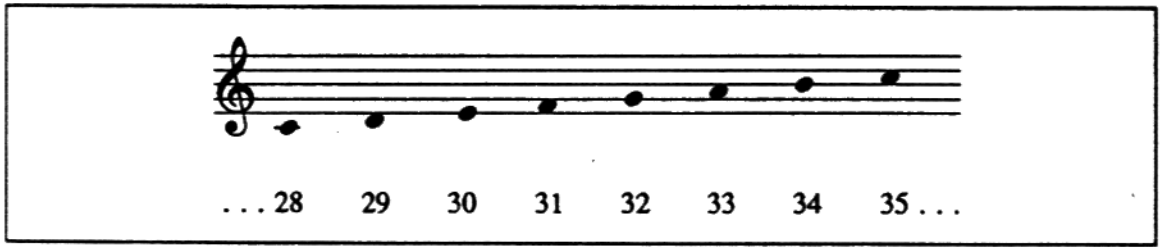
Name	*ncint*	*ncint*	*Name*
prime	0	0	octave
second	1	6	seventh
third	2	5	sixth
fourth	3	4	fifth

Intervals in the Name-Class System

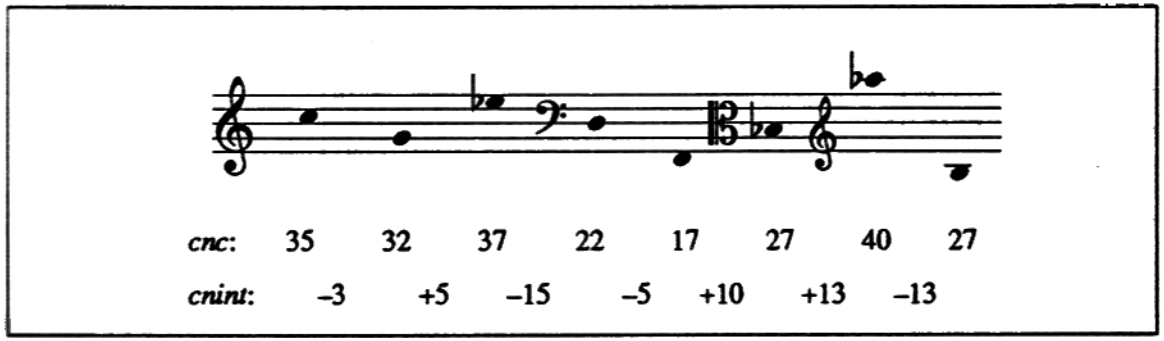


nc:	0	4	2	1	3	6	5	6
ncint:	4	5	6	2	3	6	1
nic:	3	2	1	2	3	1	1

Examples of Name-Class Integers, Intervals, and Interval Classes



The Continuous Name Code Scale



Examples of Directed Intervals in the Continuous Name Code System

binomial representation = <pc,nc>

pc equivalence: C♯ = <1,0>; D♭ = <1,1>; E♭♭♭ = < 1,2>

nc equivalence: C = <0,0>; C♯ = <1,0>; C♭ = <11,0>

Binomial Representation of Pitch

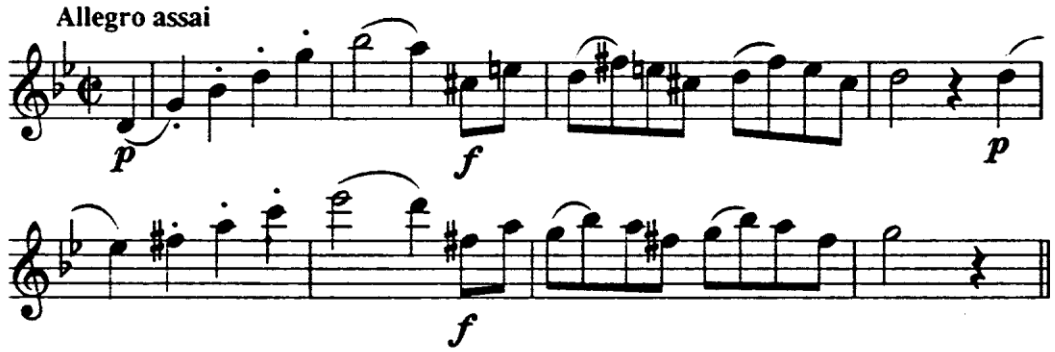
Pitch Class	*Name Class* 0	1	2	3	4	5	6
0	C	D♭♭					B♯
1	C♯	D♭					B×
2	C×	D	E♭♭				
3		D♯	E♭	F♭♭			
4		D×	E	F♭			
5			E♯	F	G♭♭		
6			E×	F♯	G♭		
7				F×	G	A♭♭	
8					G♯	A♭	
9					G×	A	B♭♭
10	C♭♭					A♯	B♭
11	C♭					A×	B

Binomial Representation of Common Notes

Pitch Class	*Name Class* 0	1	2	3	4	5	6
0	P1	d2					A7
1	A1	m2					
2		M2	d3				
3		A2	m3				
4			M3	d4			
5			A3	P4			
6				A4	d5		
7					P5	d6	
8					A5	m6	
9						M6	d7
10						A6	m7
11	d1						M7

Binomial Representation of Common Intervals

Mozart, Symphony No. 40 in G Minor, K. 550, 4th movement



Continuous Pitch Code (cpc):
50 0 55 58 62 67 0 70 69 61 64 0 62 66 64 61 62 66 64 61 0 62 99 62 0 63 66 69 72 0 75 74 66 69 0 67 70 69 66 67 70 69 66 0 67 99 100

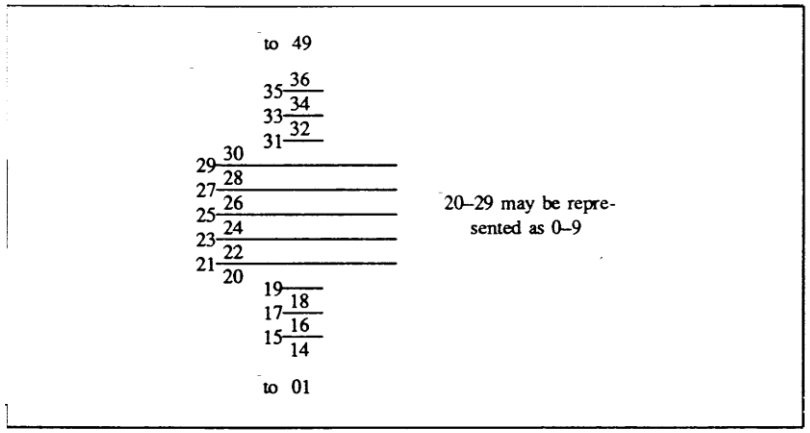
Continuous Name Code (cnc):
29 0 32 34 36 39 0 41 40 35 37 0 36 38 37 35 36 38 37 35 0 36 99 36 0 37 38 40 42 0 44 43 38 40 0 39 41 40 38 39 41 40 38 0 39 99 100

Continuous Binomial Representation (cbr):
4021 / 4074 4106 5021 5074 / 5106 5095 5010 5042 / 5021 5063 5042 5010 5021 5063 5042 5010 / 5021 r 5021 / 5032 5063 5095 6000 / 6032 6021 5063 5095 / 5074 5106 5095 5063 5074 5106 5095 5063 / 5074 r //

Reciprocal Duration Code (rdc):
4 0 4 4 4 4 0 2 4 8 8 0 8 8 8 8 8 8 8 8 0 2 –4 4 0 4 4 4 4 0 2 4 8 8 0 8 8 8 8 8 8 8 8 0 2 –4 100

DARMS Code:
!G !K2– !M$ 20QL,VP / 3 5' 7' 30' / 32HL 31Q (6#,VF 8*) / (7L 9# 8* 6) (7L 9 8 6) / 7H RQ 7QL,VP / 8– 9#' 31' 33' / 35HL 34Q (9#,VF 31) / (30L 32 31 9#) (30L 32 31 9#) / 30H RQ //

⇒ Encoding of a Simple Excerpt



DARMS Space Codes

Accidentals:

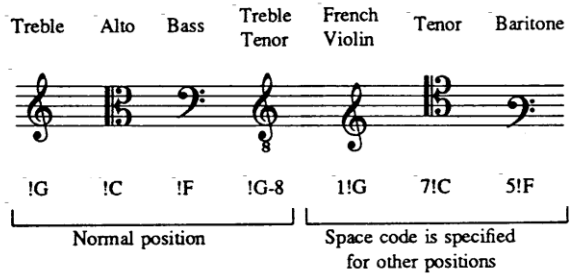
#	sharp
##	double sharp
–	flat
––	double flat
*	natural

Key Signatures:

Standard: !K2# = key of two sharps
!K4– = key of four flats

Nonstandard key signatures use accidental code followed by space code: !K–25#22 (flat on 3d line and sharp on 1st space)

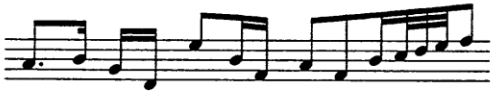
Clefs:



DARMS Encoding of Accidentals, Key Signatures, and Clefs

WWW	Longa
WW	Breve
W	Whole
H	Half
Q	Quarter
E	Eighth
S	Sixteenth
T	Thirty-second
X	Sixty-fourth
Y	One hundred twenty-eighth
Z	Two hundred fifty-sixth
ZZ	Five hundred twelfth
ZZZ	One thousand twenty-fourth
.	(dot)
G	Grace note

DARMS Duration Codes

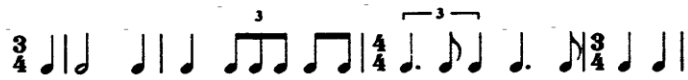


(4. (5)) ((3 0)) (8 (5 2)) (4 2 (5 (6 7 8)) 9)

Examples of Beamed Notes

!M3/4	$\frac{3}{4}$
!M6:8	$\frac{6}{8}$
!C	C
!M2+3+2:8	$\frac{2+3+2}{8}$
!M3/q.	$\frac{3}{\text{♩.}}$
!M3+2/4+1+6/8	$\frac{3+2}{4} + \frac{1+6}{8}$

Examples of DARMS Meter Signatures



Meter	3/4									1/1						3/4	
Fraction	1/4	1/2	1/4	1/4	1/12	1/12	1/12	1/8	1/8	1/4	1/12	1/6	3/8	1/8		1/4	1/4
Raw time	0/1	1/4	3/4	1/1	5/4	4/3	17/12	3/2	13/8	7/4	2/1	25/12	9/4	21/8		11/4	3/1
Fraction sum	1/4	3/4					3/4					1/1				2/4	
Part of meas.	1/3	2/3	1/3	1/3	1/9	1/9	1/9	1/6	1/6	1/4	1/12	1/6	3/8	1/8		1/3	1/3
Measure time	2/3	1/1	5/3	2/1	7/3	22/9	23/9	8/3	17/6	3/1	13/1	10/3	7/2	31/8		4/1	13/3
Beats per dur.	1/1	2/1	1/1	1/1	1/3	1/3	1/3	1/2	1/2	1/1	1/3	2/3	3/2	1/2		1/1	1/1
Beats per meas.	1/1	3/1					3/1					4/1				2/1	

Rational Duration Representation (rdr)

Q.. [♩..] = 1/4 + 1/8 + 1/16

Calculation:	*Duration*	*Dot*
duration code:	dur = 1/4	dot = 1/8
add 1st dot:	dur = 1/4 + 1/8 = 3/8	dot = 1/16
add 2d dot:	dur = 3/8 + 1/16 = 7/16	dot = 1/32
final value:	7/16	

Converting DARMS Duration Codes to RDR

DARMS code:		(	(	)	(	(	)	)	)
Parenthesis level:	0	1	2	1	2	3	2	1	0
Default duration:	1/4	1/8	1/16	1/8	1/16	1/32	1/16	1/8	1/4

Processing DARMS Beam Codes (short form)

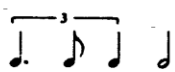
Notation	*Code*	*Meaning*
♩.	4.	dotted quarter
♩..	4..	double-dotted quarter
𝅗𝅥.	2.	dotted half
𝅘𝅥𝅯...	16...	triple-dotted sixteenth

Dotted Notes in Reciprocal Duration Code

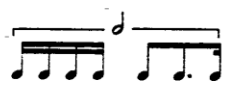
Notation	*Code*	*Meaning*
♩ ♪.	4,8.	quarter tied to dotted eight
𝅗𝅥 𝅗𝅥 𝅗𝅥	2,2,2	three tied half notes

Tied Notes in Reciprocal Durtion Code

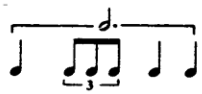
4, (4 = 8 8 8), 4



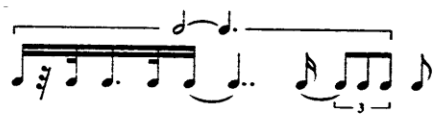
(2 = 4. 8 4) 2



(2 = 16 16 16 16 8 8. 16)



(2. = 4 (4 = 8 8 8) 4 4)



(2,4. = (4 = 16 –32 32 16. 32 16), 4.. 16, (4 = 8 8 8)) 8

Examples of Grouplet Notation in Reciprocal Duration Code

a) Grouplet identifier: !3Q7:H [♩ ♩ ♩]

b) Grouplet[7] = 1/2 + 3/4

= 1/2 × 4/3

= **2/3** [grouplet constant 7]

c) Notation: ♩ ♩. ♪

DARMS code:	Q7	Q.7	E7
Calculation:	1/4 × 2/3	3/8 × 2/3	1/8 × 2/3
Duration:	1/6	1/4	1/12

Processing DARMS Grouplets

$!n_1 dur_1 id_1 : n_2 dur_2 id_2$

where n_1 and n_2 are repetition counts,
dur_1 and dur_2 are duration codes, and
id_1 and id_2 are grouplet identifier numbers

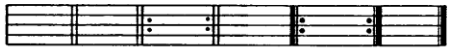
n_2 may be suppressed if $n_2 = 1$
dur_2 may be suppressed if $dur_2 = dur_1$

id_2 is used only for nested grouplet,
where id_2 is the identifier number
for a previously defined grouplet

Examples:

!3E1:Q	(3 eighths in the time of a quarter)
!7E3:2Q	(7 eighths in the time of 2 quarters)
!5Q4:4	(5 quarters in the time of 4)

DARMS Grouplet Definers



/ // //: :// !//: :/!/ /!/

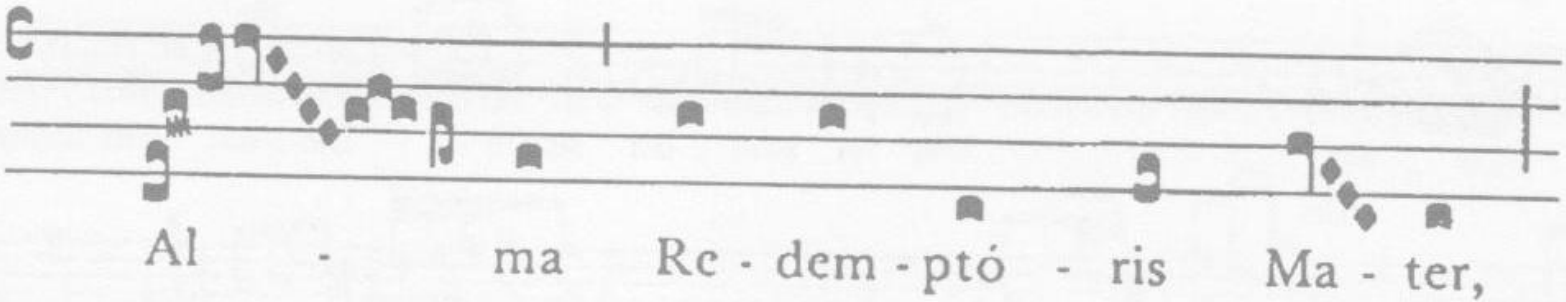
DARMS Encoding of Bar Lines

Open beams, short form [(]
Space code
Accidentals
(Note-head type)
Duration code
Tie
Stem direction
(Tremolo)
(Beam codes, long form)

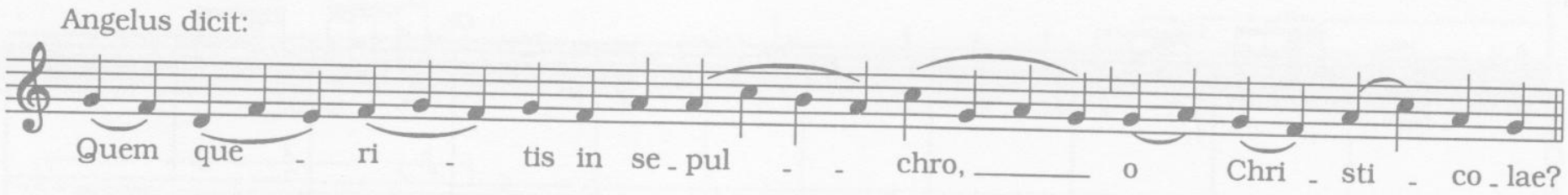
Articulation codes
Dynamics and other dictionary codes
Ornaments
Slur
(Fingering)
} ordered by relative proximity to note head

(Figured bass)
Close beams, short form [)]

DARMS Encoding Order for Notes



Example 1.5A Chant: "Alma Redemptoris Mater" written in neumes.



Example 1.5B Chant: "Quem queritis" written in modern notation.

4WJ / 4J / 4J / 4J / 4 //

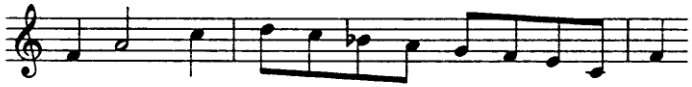
or

24WJ1 / 24WJ2J3 / 24WJ4J5 / 24WJ6J7 / 24WJ8 //

or

4WJ1 / WJ2J1 / WJ2J1 / WJ2J1 / WJ2 //

DARMS Encoding of Ties



!G 2QU 4HU 6QD / (7D 6D 5–D 4D) (3U 2U 1U 19U) / 21QU

DARMS Encoding of Stem Direction

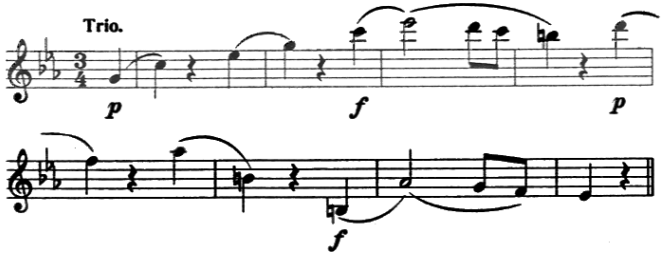
Graphic Symbol	*DARMS Code*	*Name*
⊓	?D	downbow
V	?V	upbow
φ	?Q	snap pizzicato
arco	?R	arco
pizz.	?Z	pizzicato
∘	?O	open (horn, etc.) or harmonic (string)
+	?+	stopped (horn, etc.)
H⌐	?H	Hauptstimme
N⌐	?N	Nebenstimme
¬	?⌐	terminates ?H or ?N

Implemented Dictionary Codes

Graphic Symbol	*DARMS Code*	*Name*
.	'	staccato mark
▼ or ▲	"	wedge accent
–	_	tenuto mark
>	>	accent
V or Λ	<	vertical accent
𝄐	;	fermata

DARMS Encoding of Articulation

Mozart, String Quartet No. 19, K. 465, third movement, mm. 64–71.



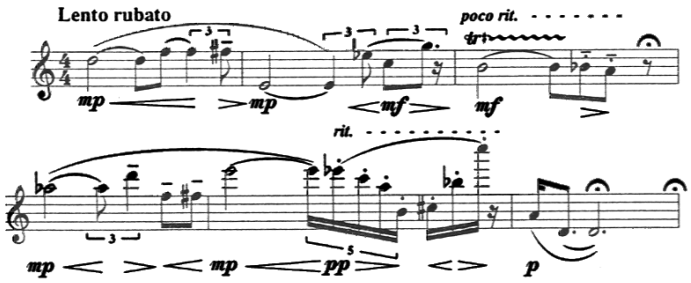
!G !K3– !M3:4 3QL / 6 RQ 8L / 30 R 33L / 35HL1 (34 33) / 32*QL2
RQ 34QL / 9 RQ 31QL / 5* RQ 18*L / 4HL1 (3 2L2) / 1Q RQ /

DARMS Encoding of Slurs and Phrase Marks

Graphic Symbol	*DARMS Code*	*Meaning*
tr	OT	trill
tr~~~~~~	O;	trill with wavy line
~	OM	mordent
∽	O?	turn
tr♯~~~~~~	O;AE#	trill with sharp on right
(♮) tr~~~~~~	O;AN(*)	trill with parenthetic natural above
♭∽♮	O?AS*AN–	turn with natural below and flat above

DARMS Encoding of Ornaments

Kazuo Fukushima, *Requiem per flauto solo*, mm. 1–6.

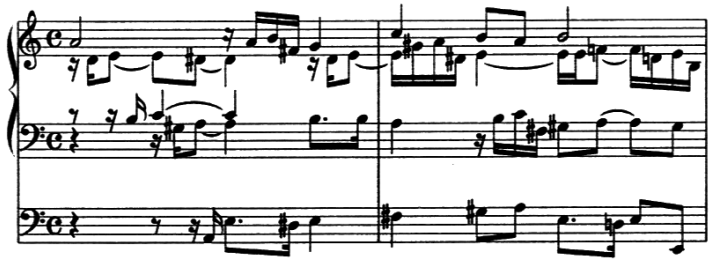


Copyright © 1966, Edizioni Suvini Zerboni, Milan. Used by permission.

```
!3E3:Q !5S5:Q
!G !M4:4,00@Lento Rubato$ 7HJL1,VMP<1 (7 9J) Q3,<2 9#E3_,V> /
1HJ,VMP Q3L2 9–E3L1,V< (6E3,VMF>1 30E3.L2,V>2) RS3 / 5HJO;AE#,VMF
(5D 5–D'_,V>1 4D'_,V>2) RE; / 31–HJL1,VMP<1 E3,V<2 34Q3_,V> (9_,V<1
9#_,V<2 / 35HJ,VMP<1 ((S5L2,V<2 35–'L1,VPP 33',V>1 30' 5',V<2))
((6#',V<1 32–',V<2>1 38'L2,V>2)) RS / ((4–L1,VP) 0.J) H.L2; /,;
```

Examples of DARMS Encoding of Dynamics

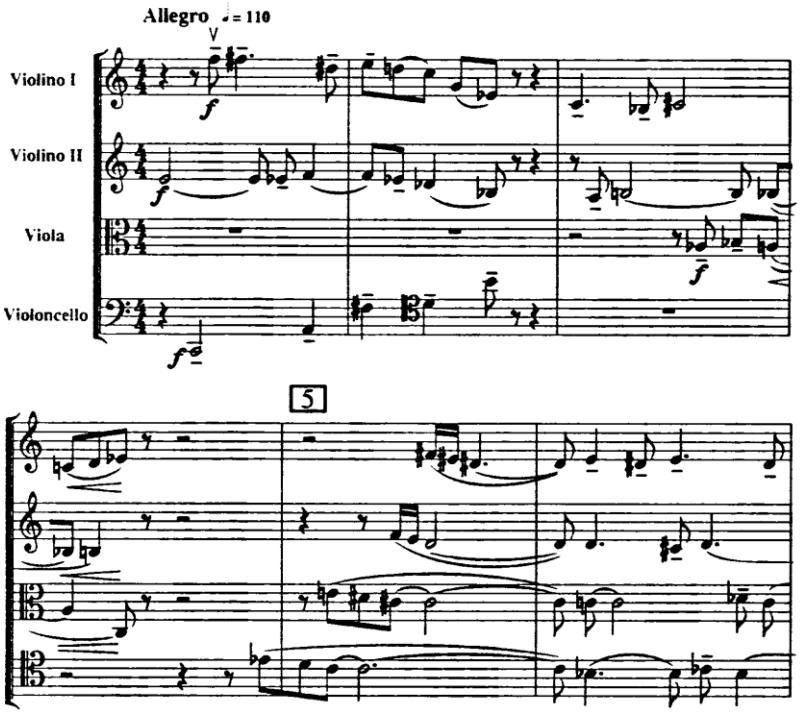
J. S. Bach, *Nun komm' der Heiden Heiland*, BWV 599, mm. 1–2.



```
!& !G !MC 4HU RS (4U 5U 2#U) 3QU / 6U (5U 4U) 5HU /
& RS ((0D) 1JD) (1D 0#JD) 0QD RSD ((0D) 1JD) /
    (1D 3#D 4D 0#D) 1QJD ((1D 1D) 2*JD) ((2D 0*D 1D 18D)) /
& !+50 !F !MC RES 30SU 31QJU QU
& RQS ((8#D) 9JD) 9QD (30.D (30D)) /
    29QD RS ((30D 31D 7#D)) (8#D 9JD) (9D 8D)
& !+100 !F !MC RQES 2S (6. (5#)) 6Q / 7#Q (8# 9) (6. (5*))
    (6 19) / &$ !+0
```

Encoding Multiple-Staff Music

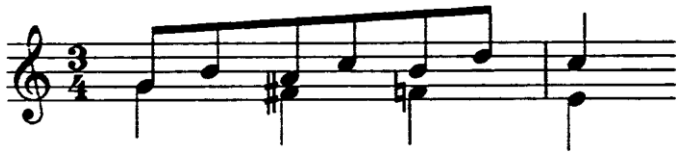
Béla Bartók, String Quartet No. 4, first movement, mm. 1–6.



Copyright © 1929, Universal Edition. Copyright Renewed, 1956.
Copyright and Renewal assigned to Boosey & Hawkes, Inc. for the U.S.A.
Used by permission.

```
I0  00@Allegro, ♩QU♩ = 110$
I1  !G  !M4:4  RQ  RE  9E_?V,VF  9#Q._  7#E_  /
I2  !G  !M4:4  1HJ,VF  E  1–E_  2QJ  /
I3  !C  !M4:4  RW  /  RW  /
I4  !F  !M4:4  RQ  17H_,VF  2Q_  /
I1  (8_  7*L  6)  (3L  1–)  REQ  /
I2  (2  1–_)  20–QL  18–E  REQ  /
I4  7#Q_  27!C  8Q_  33E_  REQ  /
I1  19Q._  18–E_  19#H  /  (19*L5,V<1  20  21–L6,V<2)  REH  /
      RH  ((2#L3  1#))  20#Q.J  /  EL4  1Q_  20#E_  1Q._  20E_  /
I2  RE  17E_  18*HJ  E  (18–JL5  /  18–,V<1)  18*QL6,V<2  REH  /
      RQE  ((2L1  1))  20HJ  /  EL2  Q.  19#E_  20QJ  /
I3  RHE  3–E_,VF  (4–_  3*JL7,V<3)  /  3Q  18EL8,V<4  REH  /
      RE  (7*L1  6#  5#J)  HJ  /  EL2  5*EJ  H  6–E_  5EJ  /
I4  RW  /  RHQE  (9–L1  /  8  7J)  H.J  /  EL2  6–Q.J  E  7–_  6QJ  /
```

DARMS Code for a Bartók Excerpt



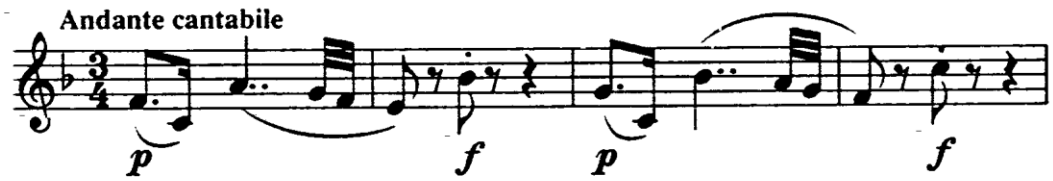
!G !M3:4 !& (3U 5U 4U 6U 5U 7U) / 6QU
& 3QD 2#D 2*D / 1D &$

Linear Decomposition Mode

!G !& 6H.U 5QU / 6HU & ¯H 0QD (3D 2D) / 1HD &$

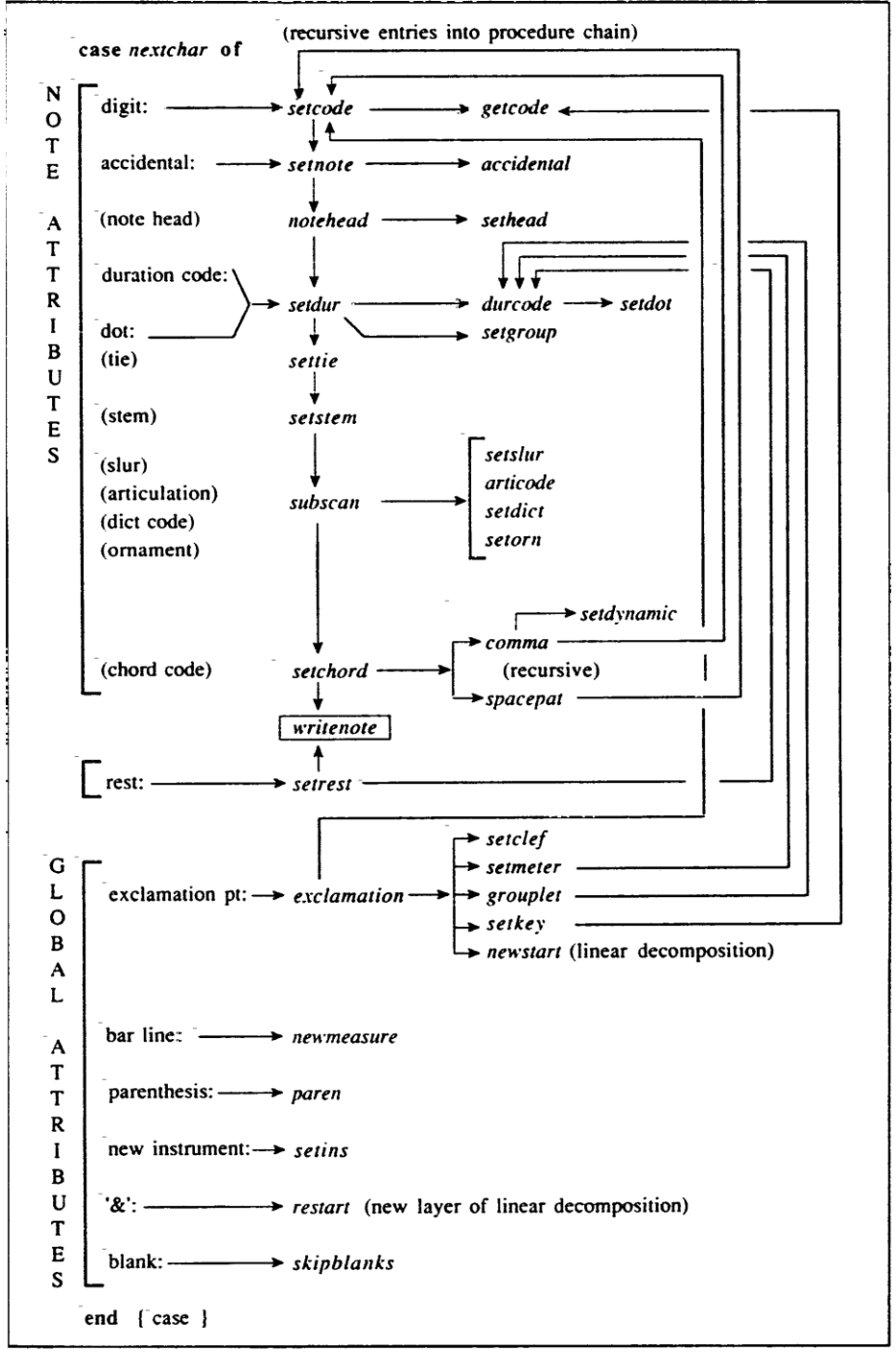
Linear Decomposition with Skip Codes

Mozart, Symphony No. 41 in C Major (*Jupiter*), K. 551, third movement




```
tempo 12 60;
i1 0 12;
   p3 rhythm 8./16/4../32//8/–8/8/–8/–4;
   p4 notes f4/c/a/g/f/e/r/bf/r/r/g/c/bf/a/g/f/r/c5/r/r;
   p5 numbers 1000*7/10000///;
end;
```

An Example of SCORE-11 Code



Program Structure for DARMS Interpreter

E = element	*ord(E)*	*Meaning*
trill	0	trill
mordent	1	mordent
turn	2	turn
flatN	3	flat above ornament
flatE	4	flat to right of ornament
flatS	5	flat below ornament
natN	6	natural above ornament
natE	7	natural to right of ornament
natS	8	natural below ornament
sharpN	9	sharp above ornament
sharpE	a	sharp to right of ornament
sharpS	b	sharp below ornament

Encoding Ornaments

E = element	*ord(E)*	*Meaning*
pppppp	0	(dynamic levels)
ppppp	1	
pppp	2	
ppp	3	
pp	4	
p	5	
mp	6	
mf	7	
f	8	
ff	9	
fff	a	
ffff	b	
fffff	c	
ffffff	d	
sfz	e	*sfz*, *ffp*, *fffmp*, *fz*, etc.
cresc	f	crescendo over single note
startcresc	g	start of crescendo over several notes
endcresc	h	end of crescendo over several notes
incresc	i	in crescendo
crescdim	j	<> over a single note
dim	k	diminuendo over single note
startdim	l	start of diminuendo over several notes
enddim	m	end of diminuendo over several notes
indem	n	within diminuendo
dimcresc	o	>< over a single note

Encoding Dynamics

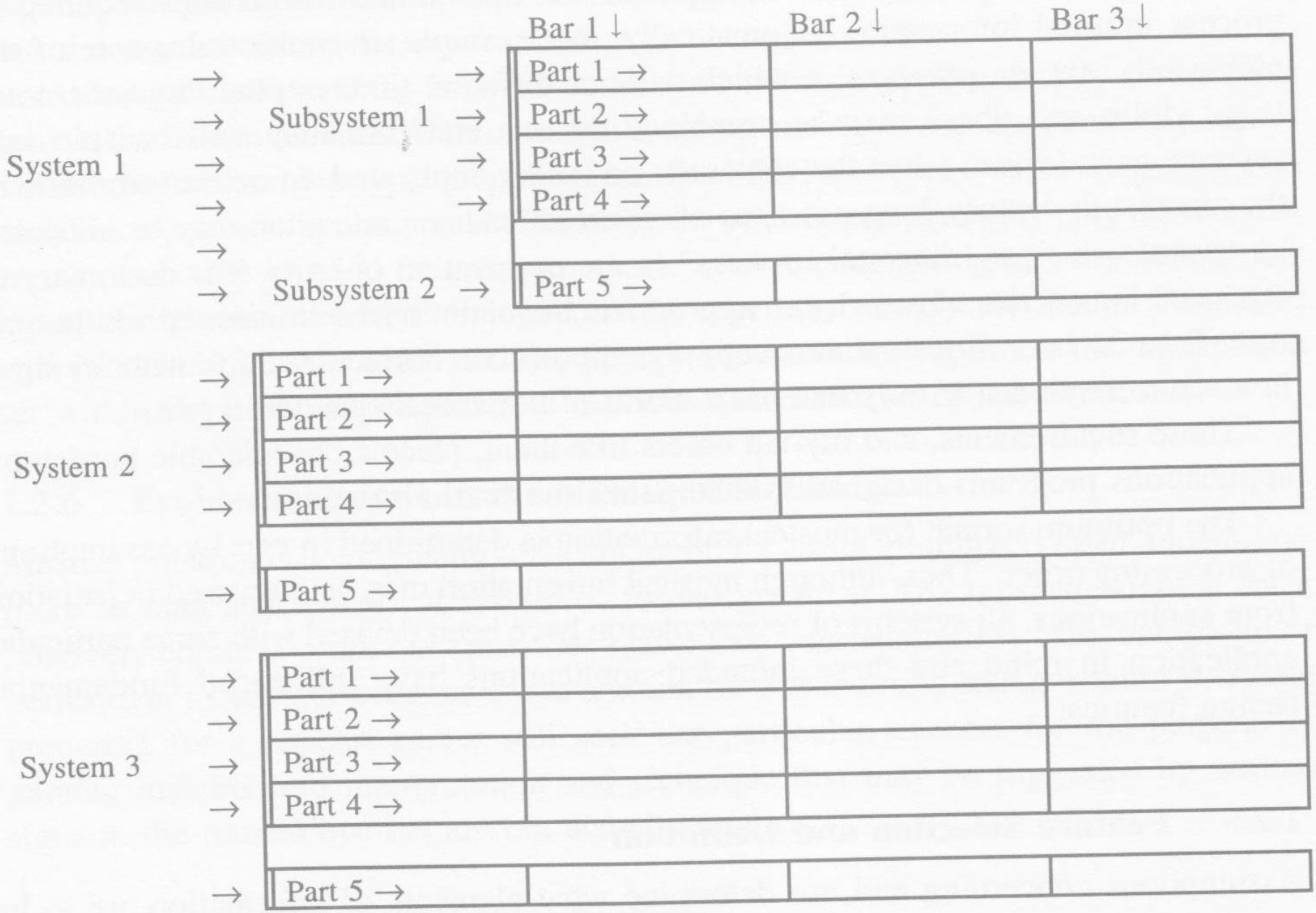
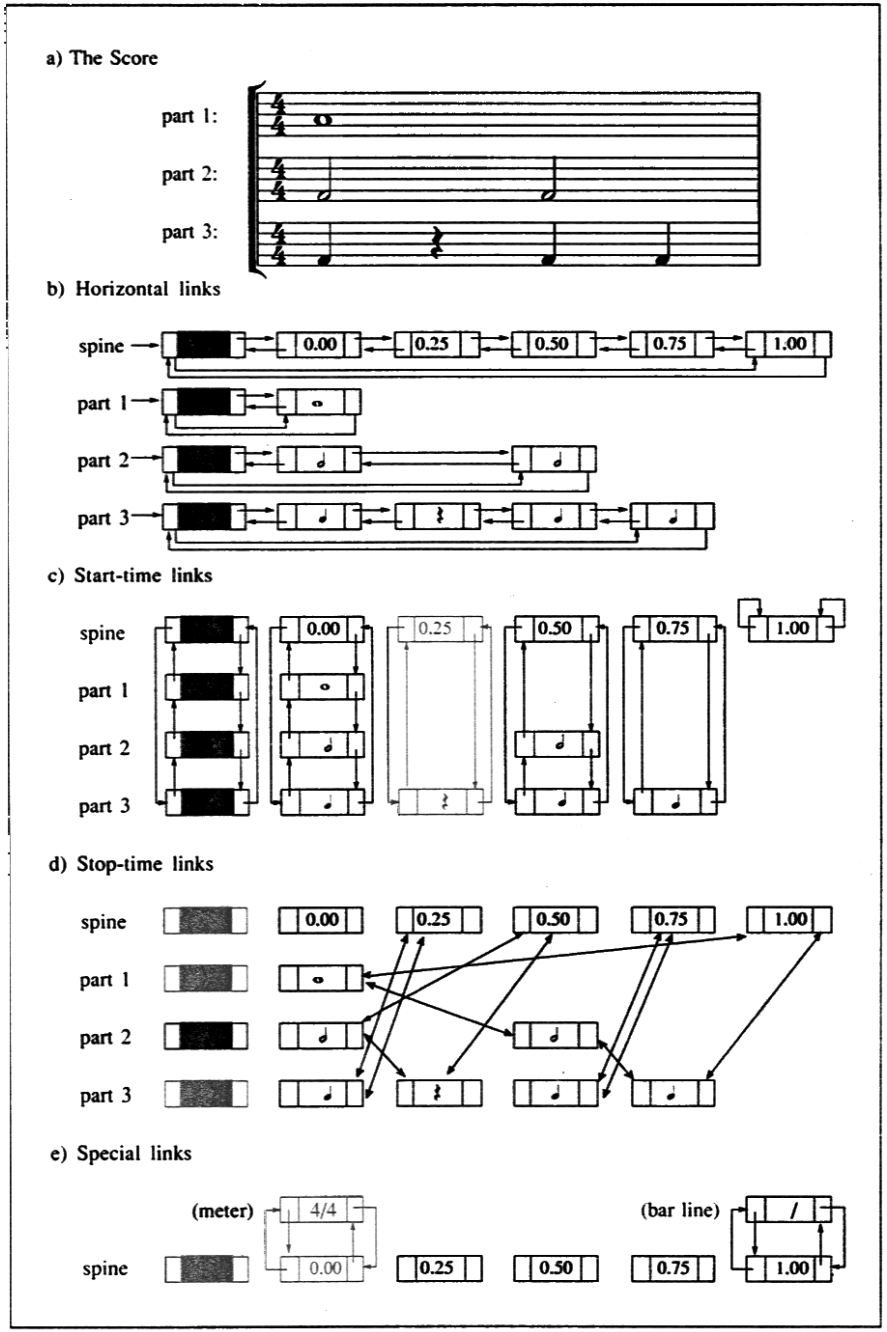
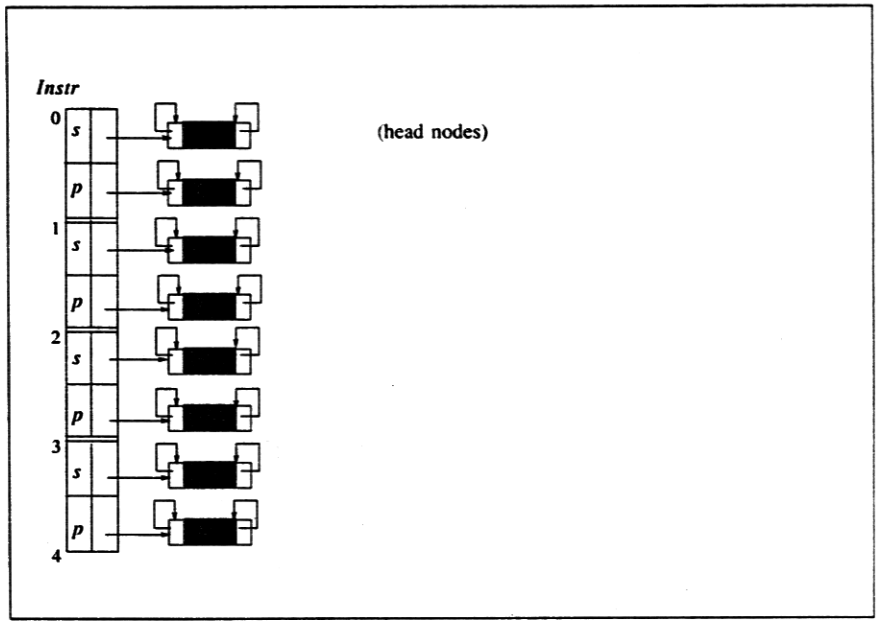


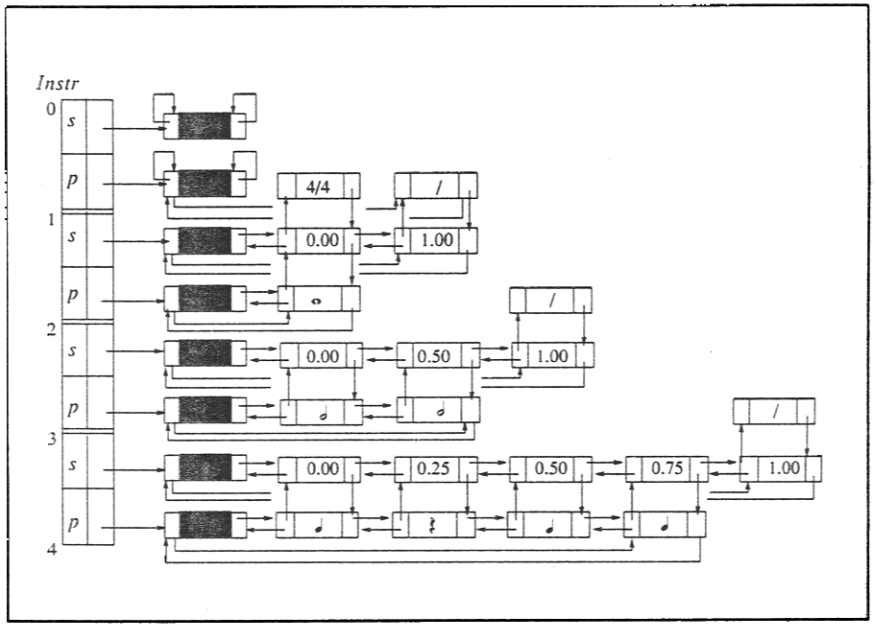
Figure 1.2 A hypothetical score viewed as a two-dimensional array.



Links in the Score Structure

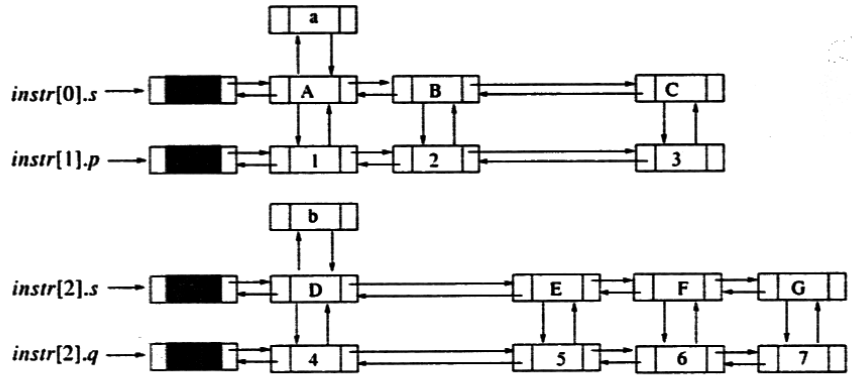


The Array *instr* after Initialization

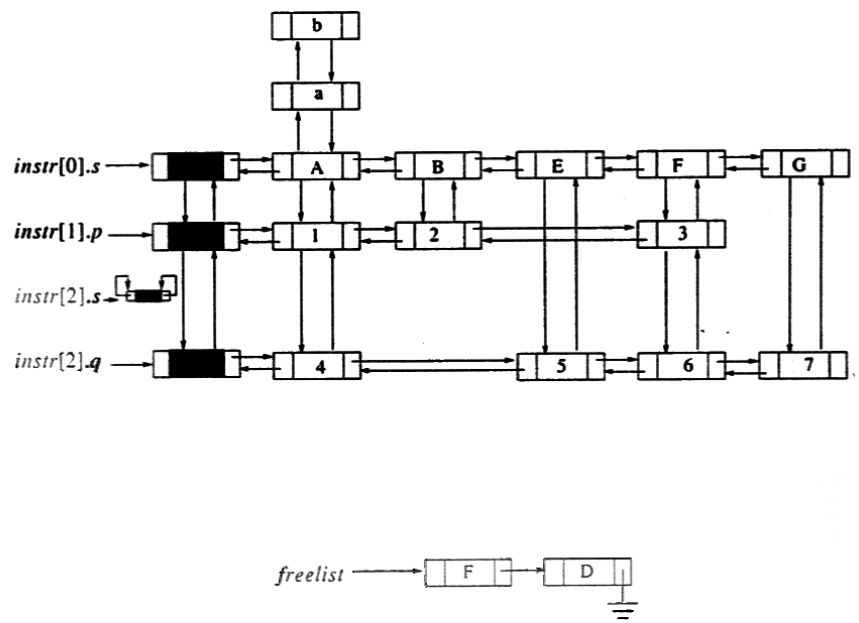


Partially Constructed Data Structure

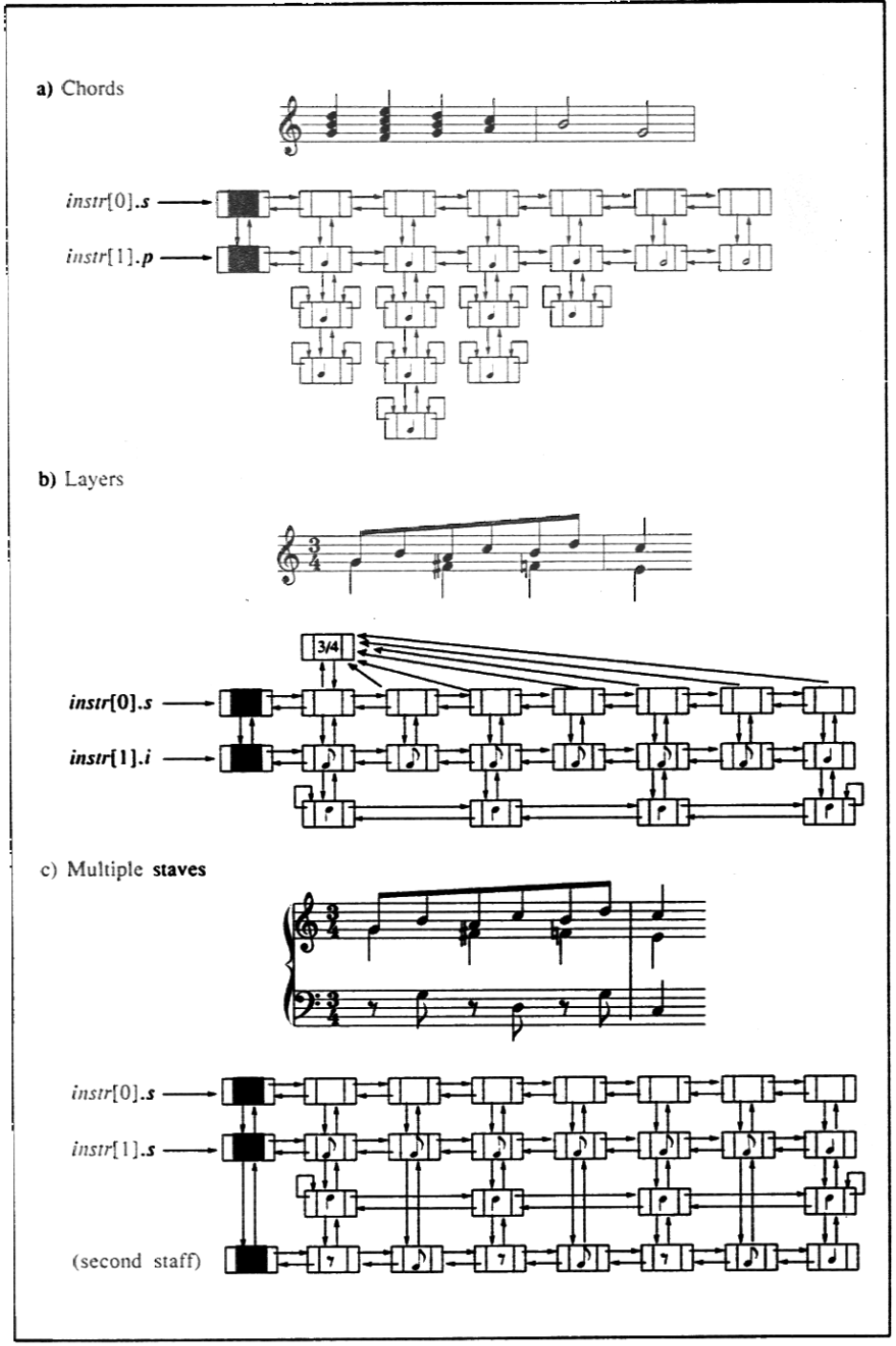
a) Before merging



b) After merging

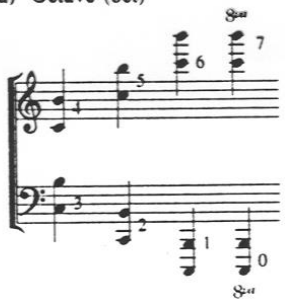


Merging Spines

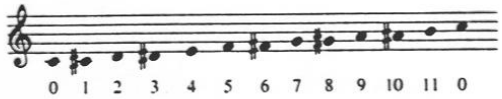


Links for Chordal Structures, Layers, and Multiple Staves

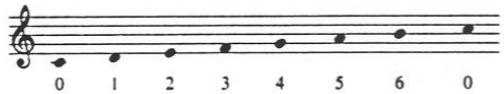
(a) Octave (oct)



(b) Pitch class (pc)



(c) Name class (nc)



(d) The continuous nc scale (cnc): $cnc = (oct \times 7) + nc$

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 . . .

octave 0 | octave 1 | octave 2 | octave 3 | . . .

(e) Unpacking the cnc scale:

$nc = cnc \bmod 7$ { remainder after division }
$oct = cnc \operatorname{div} 7$ { integer division, truncates }

(f) Mapping the space code into cnc

space code:	25	25	25
clef constant:	+9	+3	−3
cnc:	34	28	22

(g) Getting pc for natural scale from lookup table

scale

0	1	2	3	4	5	6	← name
0	2	4	5	7	9	11	← pitch class in natural scale

$pc = scale[nc]$

(h) Adjusting for key signatures and accidentals

adjust

. . .	28	29	30	31	32	33	34	35	36	37	38	39	. . .	← cnc
	0	0	0	+1	+1	0	−1	0	+2	0	+1	0		← adjustment to pc

$pc = (pc + adjust[cnc]) \bmod 12;$

(i) Packing oct, pc, and nc into a cbr:

$br = 10 \times pc + nc$
$cbr = 1000 \times oct + br$

(j) Unpacking the cbr:

$oct = cbr \operatorname{div} 1000; \quad br = cbr \bmod 1000$
$pc = br \operatorname{div} 10; \quad nc = br \bmod 10$

Decoding DARMS Pitch Codes

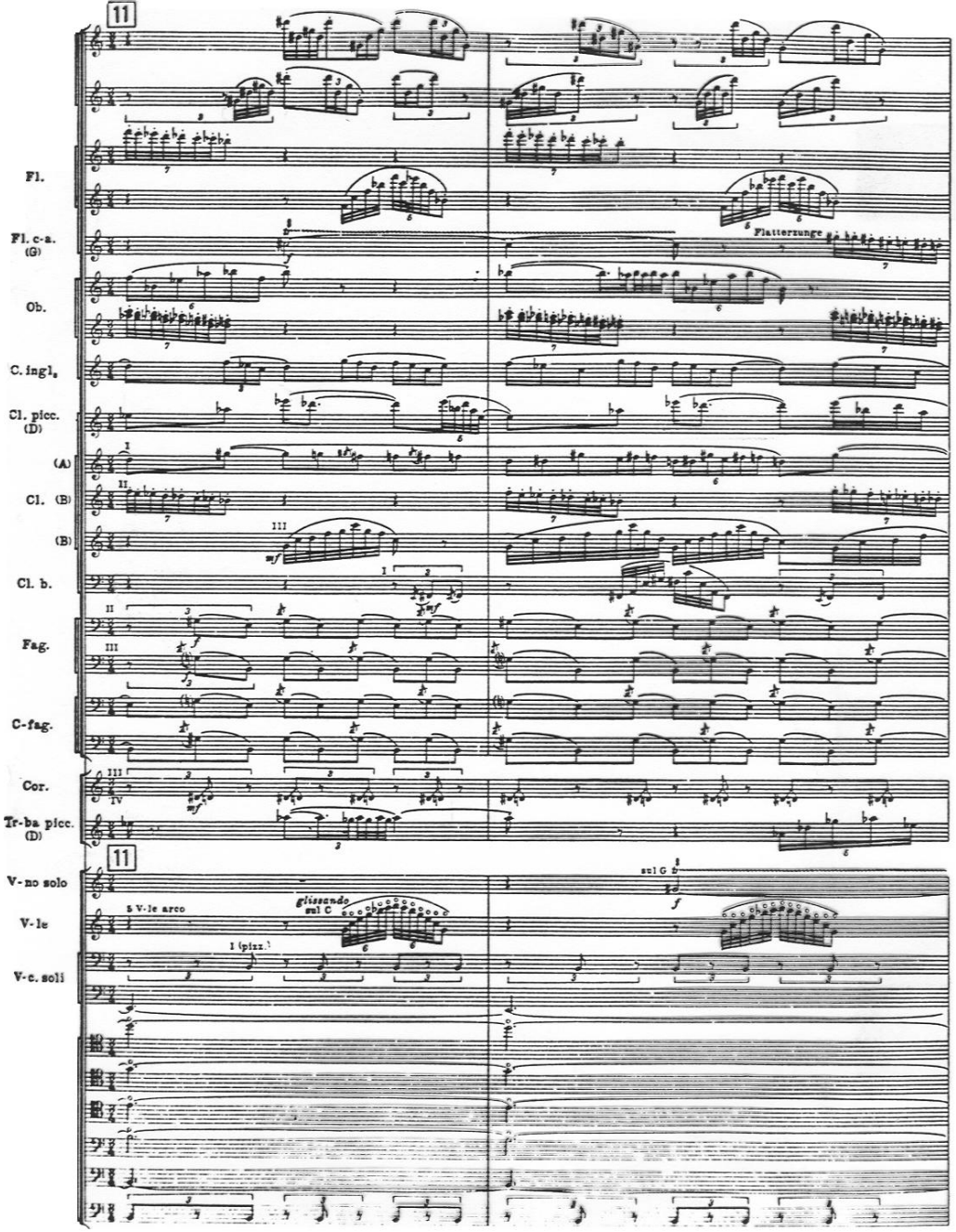


Fig. 3: I. Stravinsky, sagra della Primavera, parte prima, misure 62-63

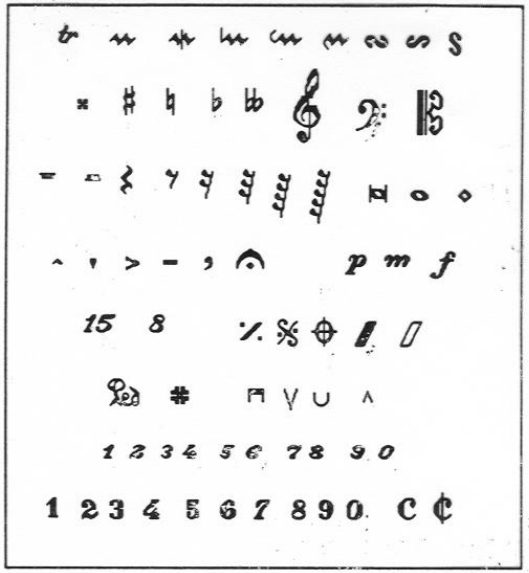
```
 1:I001(PICCOLO) 23?G !M3/4 23RQ ¬3E=Q 34#TB1(((L 27#TB1 30#TB1 27TB1)) 34EB1
 2:AN?1 ¬3S=E 21#SB1( 27S 30SB1))!L ¬0 34EB2(L2AN?1 ¬3S=E 34SB2( 27SB2
 3:AN?3E 30SB2) ¬0 23EB2)!LAN?1 / 25REAS?3 ¬3S=E 34#SB1((LASE@3E 27SB1AN@
 4:B1)!LA?1 ¬0 25REAS?2 25RE 34TB2(((LASW@3E 27TB2 30TB2 27TB2)))!L
 5: 21EB3(LAN?1 ¬3S=E 34SB3( 27SB3 30SB3) ¬0 23EB3)!LAN?1 ¬0 /
 6: I002(PICCOLO) 23?G !M3/4 ¬3E=Q 25REAS?2 25RE 23#TB1(((L 27#TB1 30#TB1 27TB1)))
 7:!L 34#EB2(LAN?1 ¬3S=E 34SB2( 27SB2 30SB2) ¬0 23EB2)!LAN?1
 8: ¬3S=E 34SB3((LAS?2 27SB3 30SB3) ¬0 34EB3)!L 25RE /
 9: 23*TB1(((L1AS?2 27#TB1 30#TB1 27TB1)) 34#EB1)!L1AS@3E 25RE
10: 25RE AS?2 21TB2(((L2 27TB2AS 3E 30TB2 27TB2)) 34EB2)!L2
11: ¬3S=E 23SB3L3AS?2 27SB3 30SB3) ¬0 34EB3)!L3AS@3E 25RE ¬0 /
12: I003(F.L) 23?G !M3/4 ¬7T=E 36TBAN?1 35-TB?1 34TBAN?1 34-TBAN?1.ASW@7E
13: 33TB?1 33-TB))AN?1 32EB)AN?1 ¬0 25RQ 25RQ /
14: ¬71=E 36TBAN?1 35-TB?1 34TBAN?1 34-TBAN?1.ASW@7E
15: 33TB?1 33-TB))AN?1 32EB)AN?1 ¬0 25RQ 25RQ /
16: I004(F.L) 23?G !M3/4 25RQ 25RE 22TB1(((L1 26TB1 29TB1 32TB1)))
17: ¬51=E 35TB2(((( 33TB2 35-TB2AS@5E 32TB2 29TB2)) ¬0 25-EB2)!L1AN?1 /
18: 25RQ 25RE ¬5T=E 22TB1(((L1 26TB1 29TB1AS@5E 32-TB1 35-TB1)))
19: 35TB2(((( 33TB2 35TB2AS@5E 32TB2 29TB2)) ¬0 25-EB2)!L1AN?1 /
20: I005(F.L.C&A) 23?G !M3/4 25RQ 26#HJU1OUT1AN@#E.VF / 26!JU1QJU2
21: 25!JU2EAN?7 25RE ¬7T=E 30#TB(((ANW@FLATTERZUNGEEAN?1 30*TRAN?1
22: 29*TRAN?1 28*TRAN?1AS@7E 28*TBAN?1 27#TRAN?1 27*TB)))AN?1 ¬0 /
23: I006(UB) 23?. !M3/4 ¬6S=Q 29SB(((L 25-SB 28-SBASE@6E 33-SB 34-SB 29SB)) ¬0
24: 32E!LAN?1 25RF 25RQ / 32-EJU1B1(L1 32!JU1S.B1( B1( 31-TB1)) 32TB1(((
25: 31TB1 32TB1 31TB1))) ¬6S=Q 32SB2(( 25-SB2 28-SB2AS@6E 31SB2 32SB2 29SB2))
26: 25S!L1 25RE. /
27: I007(UB) 23?G !M3/4 ¬7T=E 33TB(((AN?1.31-TB 32TBAN?1.30TB 32-TRAN?1.30-TB
28: 31*TBAN?1.29TB 31-TBAN?1.29-TB 30*TRAN?1.28-TB 29#TB))AN?1.27-TB ¬0
29: 29*EB)AN?1.27-EB 25RQ 25RQ /
30: ¬71=E 33TB(((AN?1.31-TB 32TBAN?1.30TB 32-TRAN?1.30-TB
31: 31*TBAN?1.29TB 31-TBAN?1.29-TB 30*TRAN?1.28-TB 29#TB))AN?1.27-TB ¬0
32: 29*EB)AN?1.27-EB 25RQ 25RE ¬7T=E 33TB2(((AN?1.31TB2 32*TB2AN?1.30*TB2
33: 32-TB2AN?1.30-TB2 31*TB2AN?1.29*TB2 31-TB2AN?1.29-TB2 30*TB2AN?1.28-TB2
34: 29*TB2)))AN?1.27TB2 ¬0 /
35: I008(C.INGL.) 23?G !M3/4 27!JUEB1( ¬3S=E 29SB1(L1 28-SB1AS@3E 26SB1)) ¬0
36: 27EB2(!L1 ¬3S=E 30SB2L2 29SB2 27SB2)) 29SB3(( 28SB3 26SB3) ¬0
37: 29EB3)!L2 / ¬3S=E 29SB1((L1 28-SB1 26SB1) 30SB1( 29SB1 27SB1)
38: 29SEB2(( 28SB2 26SB2) ¬0 27EJUB2) 27!JUEB3(!L1 ¬3S=E 29SB3(L2 28SB3 26SB3
39:)) ¬0 /
40: I009(CL.PICC.) 23?G !M3/4 26EB1( 31-FB1) 35-SB2((L1 B2 34E.JUB22)
41: 34!JUEB3(!L1 ¬5T=E 35TB3((L 32-TB3 34TB3AS@5E 31TB3 28TJUB3))) ¬0 / 28!JUEB1(!L
42: 31EB1) 35S-B2((L B2) 34-E.JUB2) 34!JUEB3(!L 35TB3(( 32-TB3 34TB3 31TB3))) /
43: I010(BA1) 23?G !M3/4 27!JUEB1(AN@IE 30#EJUB1)L ¬3E=Q 30!JUEB2( 30*EB2
44: 29*EB2)ANW@!!1 30*EO5E 29*EB2( 29#EB2ANW@!!1 30EO5E 29*EB2) ¬0 /
45: ¬6S=Q 27SB3(( 27#SB3 30#SB3 28SB3 29#SB3 29*SB3)) 27*SB4(( 27#SB4
46: 30#SB4 28SB4
47: 29*SB4 29*SB4)) ¬0 27EB5(!L 30EJUB5)L /
48: I011(SB1) 23?G !M3/4
49: ¬71=E 29TB(((ANW@IIEAN?1 28TBAN?1 28-TBAN?1 27TBAN?1 27-TBAN?1 26TBAN?1
50: 25-TB))AN?1 ¬0 25-EB)AN?1 25RQ 25RQ /
51: ¬71=E 29TB(((ANW@IIEAN?1 28TBAN?1 28-TBAN?1 27TBAN?1 27-TBAN?1 26TBAN?1
52: 25-TB))AN?1 ¬0 25-EB)AN?1 25RQ 25RE ¬7T=E 30TB1((AN?1 30-TB1AN?1 29TB1AN?1
53: 29*TB1AS@7EAN?1 28-TB1AN?1 27*TB1AN?1 27-TB1)))AN?1 ¬0 /
54: I012(CL.SB1) 23?G !M3/4 25RQ 23TB(((LAN@III£.VMF 26TB 27TB 29TB 30TB 33TB
55: 30TB 29TB))) 26E!L 25RE / 23TB(((L 26TB 27TB 29TB 30TB 33TB 30TB 27TB)))
56: 23TB1((( 26TB1 27TB1 29TB1 30TB1 33TB1 30TB1 27TB1))) 26EB2(!L 23TB2((L
57: 26TB2 27TB2 29TB2))) /
58: I013(CL.B) 27?F !M3/4 25RQ 25RQ ¬3E=Q 25REANW@IEAN?3 20#EB(AW@!!1_19EO5E
59: 20EB)AW@!!1_19EO5E ¬0 / 25RE 20#TB1(((L 22B1 26TB1 28*TB1))) 27*TB2((( 31TB2
60: 25TB2 22TB2)) 20TB2)!L ¬3E=Q 25REAN?3 20EB3(AN@3EAW@!!1_19EO5E 20EB3) ¬0 /
61: I014(FAG) 27?F !M3/4 ¬3E=Q 25REAN?3 28#EB(LAN@3E.VF 26EB)!L
```

esempio 8 a)

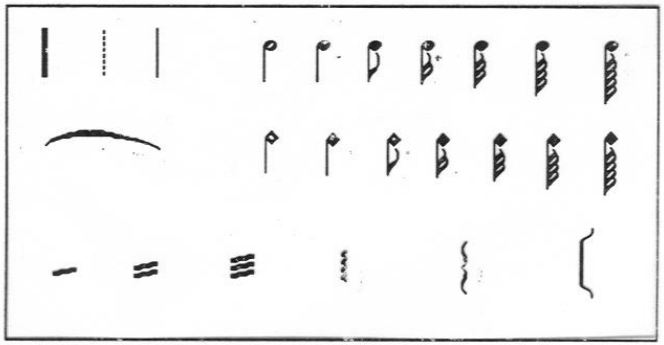
```
62: 32E05£ 26EB1!L 28ER1)L 28FB2!L 28ER2LANW@!!1_32E05£ 26EB2)!L /
63: 28*EB(L 26ER!L 28ER!LANW@!!1_30E05£
64: 26FB1(!L 28ER!L 26FB1)!L 28EB2(LANW@!!1_32E05£ 26EB2!L 28EB2)L ¬0 /
65: I015(FAG) 27?F !M3/4 ¬3E=0 25REAS?3 28ER(LANW@!!1_32E05£AS@3£,VF 23ER)!L
66: 28EB1(L 23EB1!L 28EB1)LANW@!!1_32E05£ 23EB2(!L 28EB2L
67: 23EB2)!L / 28ER(LANW@!!1_32E05£ 23EB!L 28EB)L
68: 23EB1(!L 28EB!LANW@!!1_32E05£ 23EB1)!L 28EB2(L 23EB2!L 28E
69: B2)LANW@!!1_32E05£ ¬0 /
70: I016(CFAG) 27?F !M3/4 ¬3F=0 26!JUER( 28FBL 26EB)!L 28EB1(LANW@!!
71: 1_32E05£ 26EB1!L 28EB1)L 26E 2(!L 28FB2LANW@!!1_30E05£ 26EB2)!L
72: 7 28*EB3(L 26EB3!L 28EB3)LAN @!!1_30E05£ 26EB4T!L 28EB4L
73: 26EB4)!L 28E-5(LANW@!!1_30E05£ 26EB5!L 28EB5)L ¬0 /
74: I017(C.FAG) 27?F !M3/4 ¬3F=W 26!JUER( 28*EBLANW@!!1_30E05£ 23EB)!L
75: 28EB1(L 23B1!L 28EB1)LANW@!!1_30E05£ 23FB1(!L 28EB2L 23EB2)
76: !L / 28*EB(LANW@!!1_30E05£ 23EB!L 28EB)L 23EB1(LANW@!!1_3
77: 0E05£ 23EB1)!L 28EB2(L 23EB2.L 28EB2)LANW@!!1_30E05£ ¬0 /
78: I018(COR) 23?G !M3/4ANDTIIE£AS@IV£ ¬3E=0 25REAN?3 20#EAN@3£,VMF.20*F 25RE
79: 20*EB(AN?3,.20*EB 25RE4 20#EB).20*EB 25RFAN?3 20#EAN@3£,20*F
80: 25RE 20#E B1(.20*EB1 25REB1 20#EB1).20*EB1 25RE AN?3 20#EAN@3£
81: .20*E 25RE / 20#EB1(.20*EB1 25RE 20#EB1).20*EB1 25RE 20#E.
82: 20*F 25RE 20#EB2(.20*EB2 25REB2 20#EB2).25EB2 ¬0 /
83: I019(T.HA.PICC) 23?G !M3/4 26-S 25RE. 31-EJUB(L 31!JUS.B( B( 30-TB)) 31TB((
84: 30!B 31TB 30TB))) 31QJU!L / 31!JUE 25RE 25RQ ¬5S=Q 24-SB1(( 27-SB1 30-SB1
85: 33-SB1ASW@3£ 28-SB1)) ¬0 /
86: I020(V.NO.SOLO) 23?G !M3/4ANE?4 27RQ 27RQ 27RQ / 25RQ 23#HJDOOT1ANW@SUL.G£AN@
87: #£.VF /
88: I021(V.LE) 23?G !M3/4 25RQANE@5_V.LE.ARCO£ 25REANE@GLISSANDO£ANES@..SUL.G£
89: ¬51=E 23TB4((LAN?5 26TBAN?5 28TBAN?5 30TBAS@6£AN?5 32TBAN?5 33-TB)))AN?5
90: 34TB1(((AN?5 33TB1AN?5 32TB1 N?5 30TB1AS@6£AN?5 28TB1AN?5 26TB1))AN?5
91: 23EB1)LAN?5 ¬0 / 25RQ 25RE
92: ¬61=E 23TB4((LAN?5 26TBAN?5 28TBAN?5 30TBAS@6£AN?5 32TBAN?5 33-TB)))AN?5
93: 34TB1(((AN?5 33TB1AN?5 32TB1 N?5 30TB1AS@6£AN?5 28TB1AN?5 26TB1))AN?5
94: 23EB1)LAN?5 ¬0 /
95: I022(V=C.SOLI) 27?F !M3/4 ¬3E=W 25REAS?3 25REAS@3£ 2#EAN@I(PIZZ..)£
96: 25REAS?3 21EAS@3£ 25RE 23EB(AS?3 25REAS@3£ 23EB) /
97: 25REAS?3 21EAS@3£ 25RE 23EB1(AS?3 25REB1AS@3£ 23EB1)
98: 25REAS?3 23EAS@3£ 25RE ¬0 /
99: I023(V=C.SOLI) 27?F !M3/4 19!JDH.JD / 19!JDH.JD /
100: I024(CB) 27?C !M3/4 35!JUH.JUAN?5 / 35!JUH.JUAN?5 /
101: I025(CB) 27?C !M3/4 31!JUH.JUAN?5 / 31!JUH.JUAN?5 /
102: I026(CB) 27?C !M3/4 29!JUH.JUAN?5 / 29!JUH.JUAN?5 /
103: I027(CB) 27?F !M3/4 30!JUH.JUAN?5 /30!JUH.JUAN?5 /
104: I028(CB) 27?F !M3/4 23!JDH.JD / 23!JDH.JD /
105: I029(CB) 27?F !M3/4 ¬3E=0 23EB(AN?3 25REBAN@3£ 23EB) 25REAN?3 23EAN@
106: 3£ 25RE 23EB1(AN?3 25REB1 23EB1) / 25REAN?3 23EAN@3£ 25RE
107: 23EB2(AN?3 25REB2AN@3£ 23EB2) 25REAN?3 23FAN@3£ 25RE ¬0 ///
```

esempio 8 b)

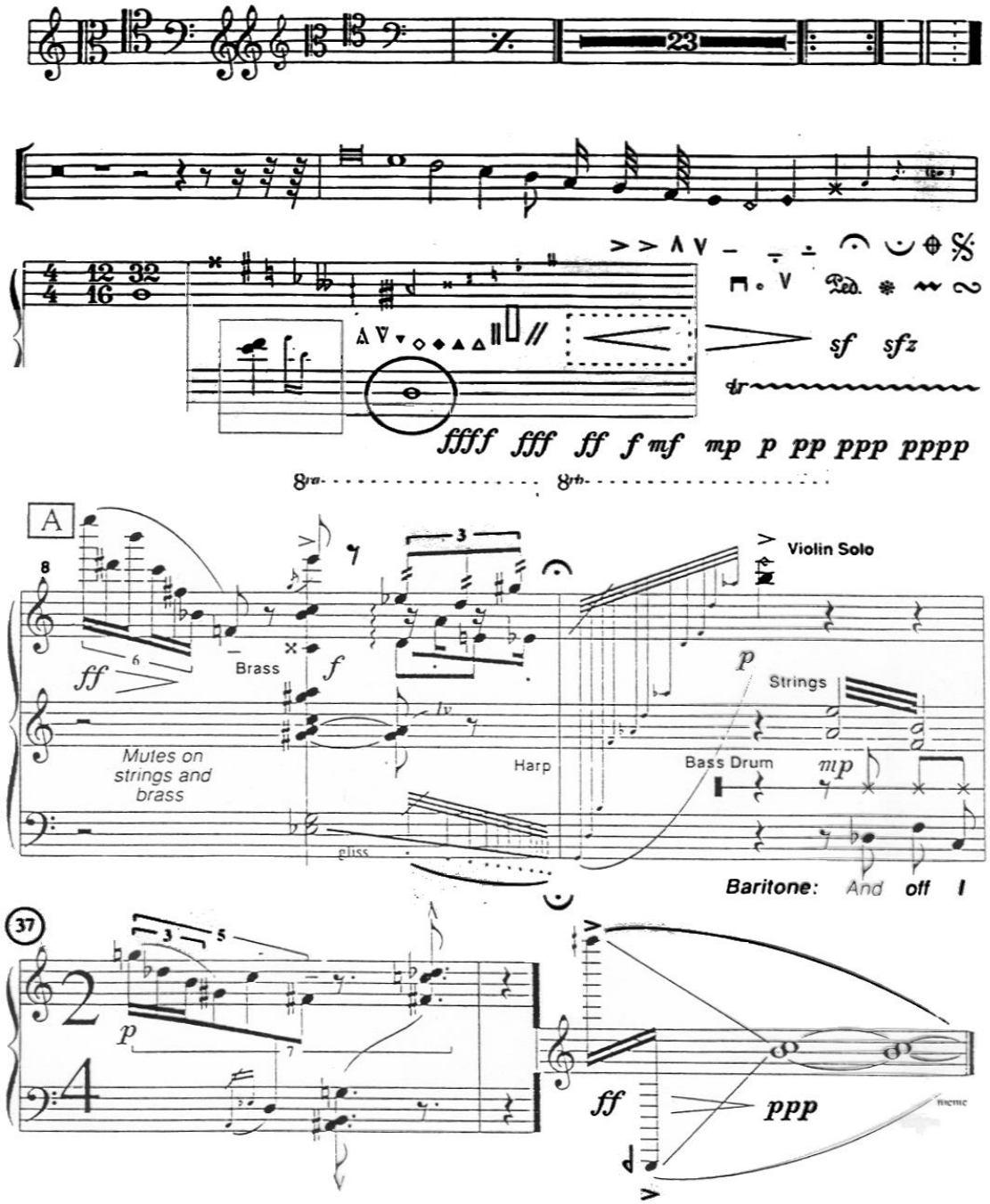
"MUSFONT"



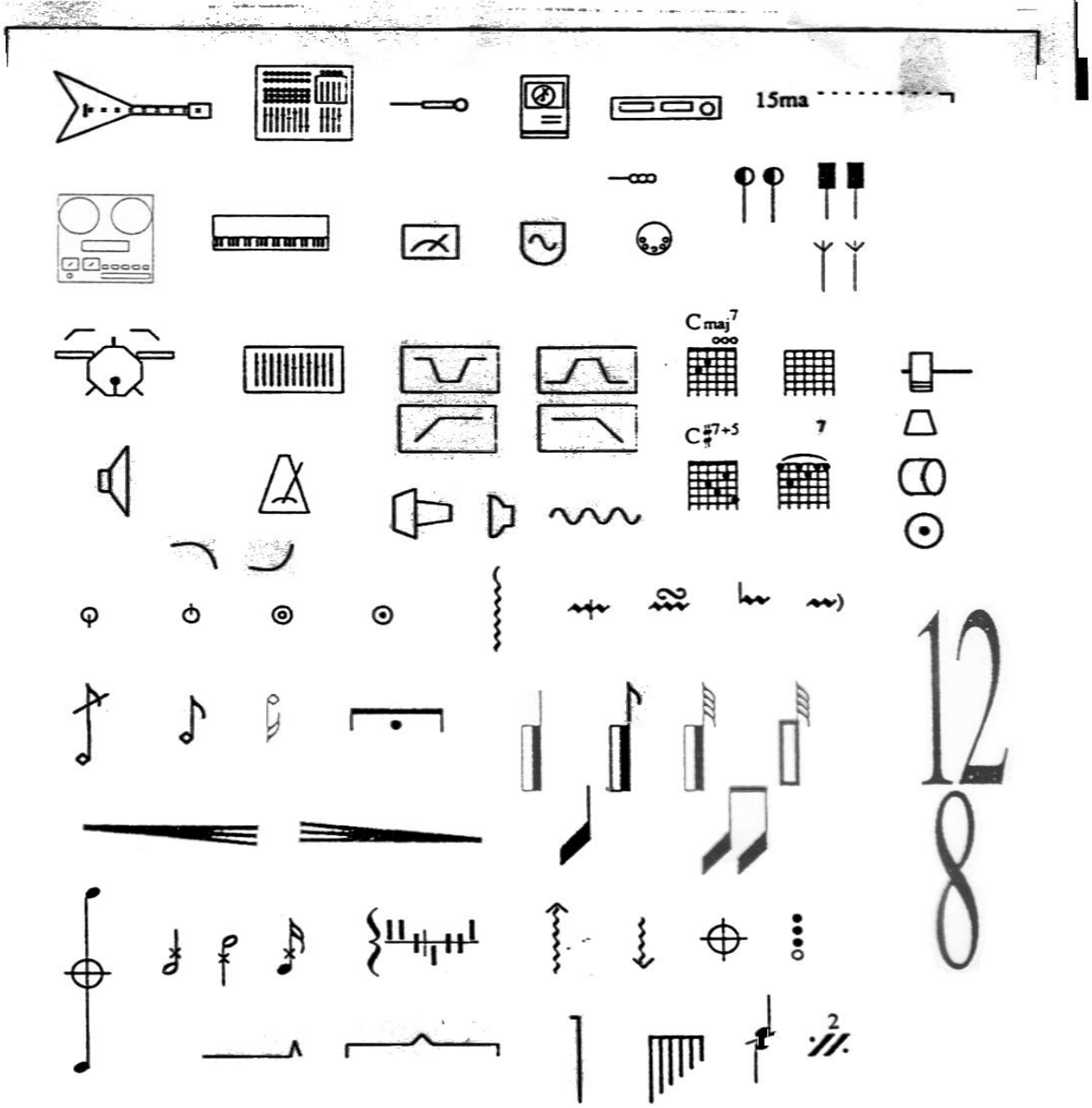
Fixed symbols



Dynamic symbols

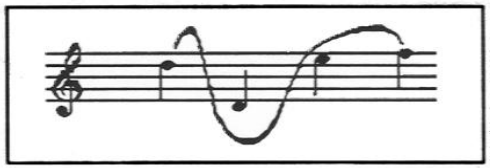


Music symbols and examples of the graphics-based notation editor



Symbols created for a notation editor library.

SPECIAL CASES



Capitolo 6

Livello strutturale

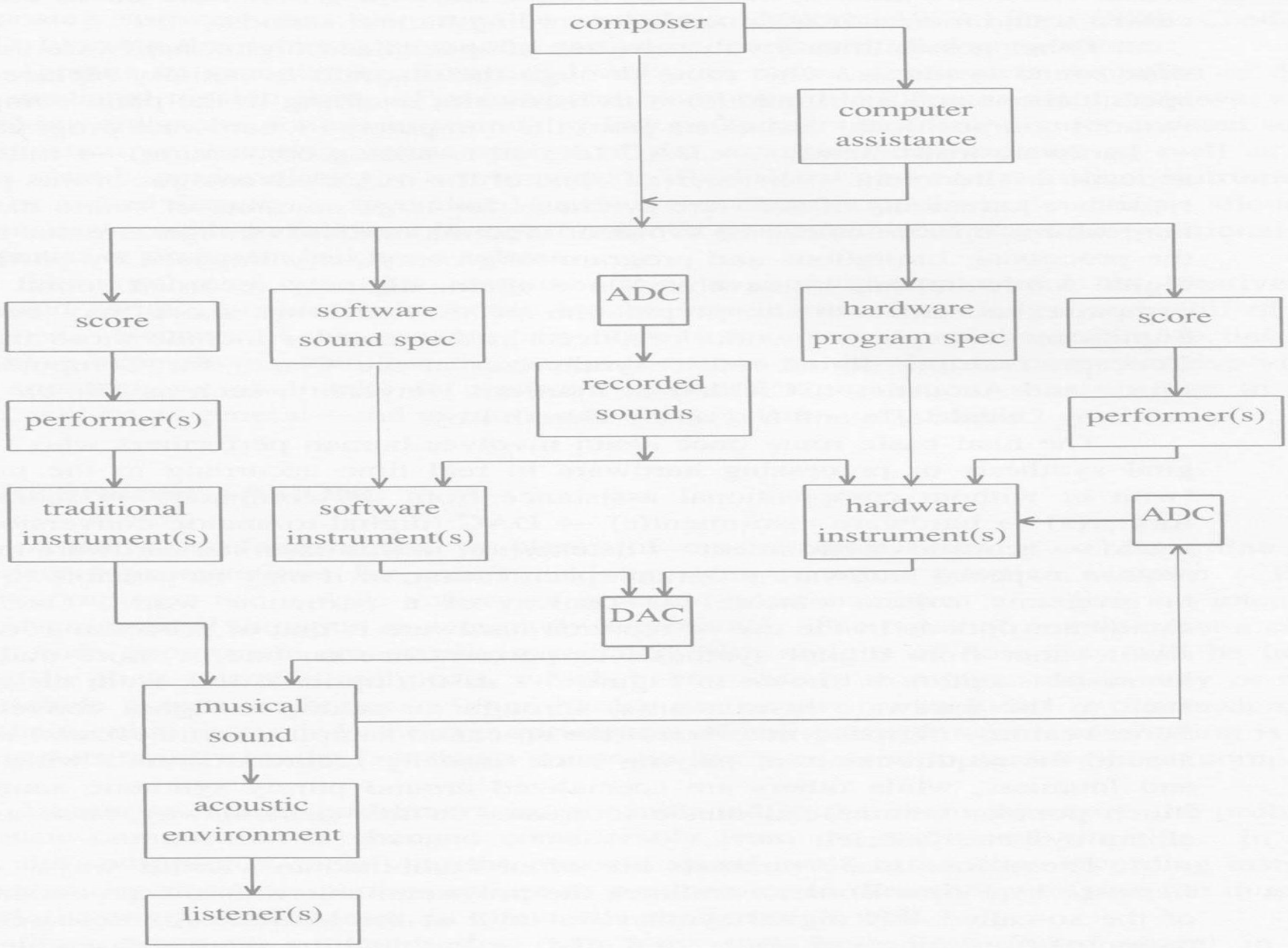


Figure 5-1 Computer-mediated music composition. The links in this diagram represent information flow. All links are optional, but one or more paths must connect the composer to the musical sound.

Capitolo 7

Descrizione dell'informazione musicale multistrato

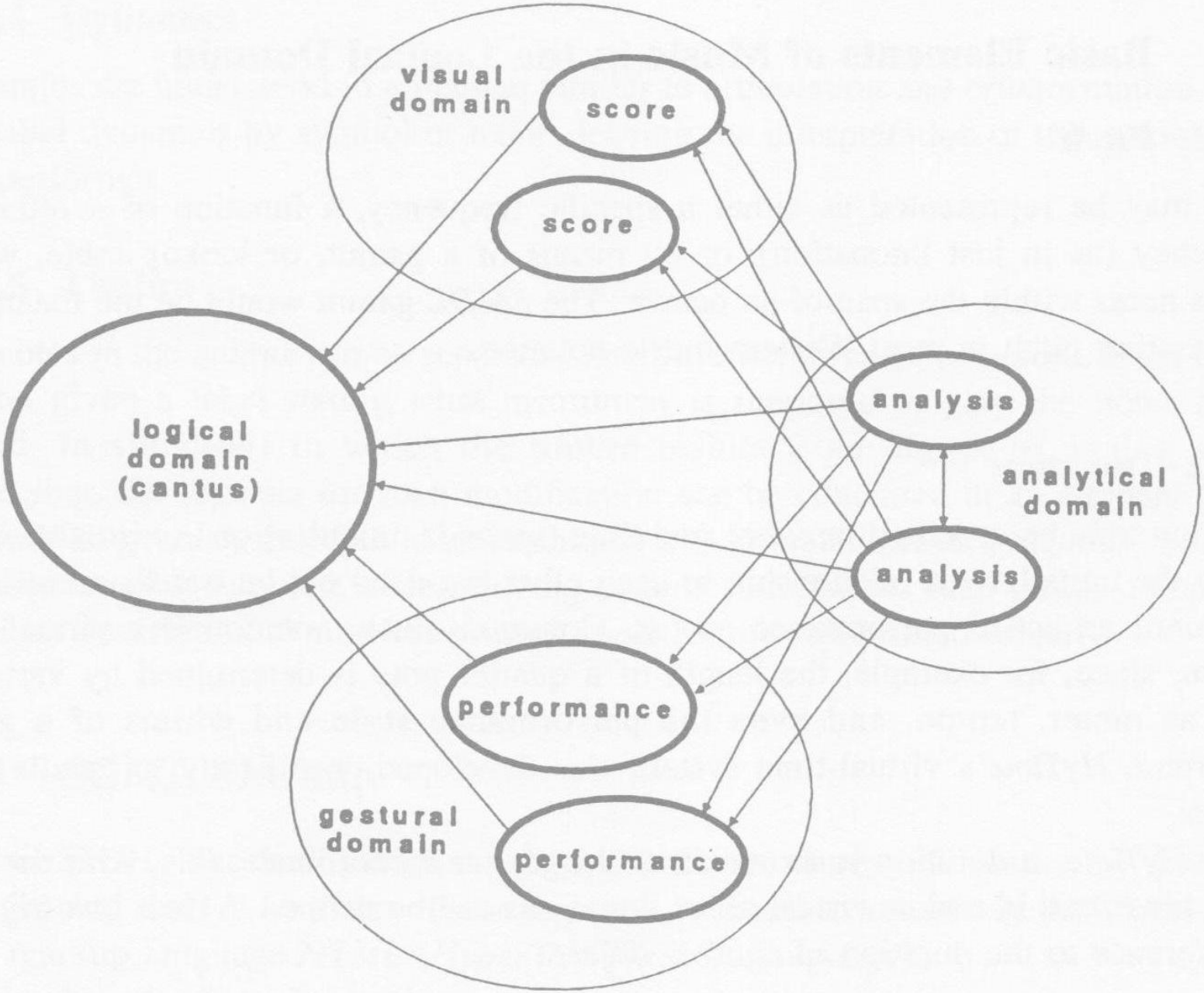


Figure 30.1 Domains of description in *SMDL*.

Capitolo 8

Tecnologie per il Digital Rights Management