2. Additive Synthesis: Context and Development *19*

2.1 Overview **19**

2.2 Alternative Synthesis Techniques **19**

2.2.1 Processed Recording 19

2.2.2 Physical Modelling 21

2.2.3 Abstract Algorithm 23

2.3 Developments in Spectral Modelling **24**

2.4 IFFT Simulation of the TOB **31**

2.4.1 Overlap-Add Synthesis 32

2.4.2 Supporting Frequency Envelopes 33

2.4.3 Noise Synthesis 34

2.4.4 Performance Evaluation 35

2.5 Review **36**

Figure 2.1 Overlap-Add IFFT Synthesis with Triangular Windows *32*

2. Additive Synthesis: Context and Development

2.1 Overview

AS is but one of many music synthesis algorithms. Smith (1991) suggests a taxonomy of four categories; (i) Processed Recording, (ii) Spectral Modelling, (iii) Physical Modelling and (iv) Abstract Algorithm. Section 2.4 is devoted to tracing the development of (ii) as it is of particular relevance to this thesis. Section 2.2 is a review of the remaining categories with a primary motive of investigating how they compare with AS in both quantitative and qualitative terms (e.g. computational efficiency, intuition of control). The most significant advance that is identified is the application of the IFFT for TOB simulation. As MAS shares the same objective, this topic is documented in detail in section 2.4.

2.2 Alternative Synthesis Techniques

2.2.1 Processed Recording

'Processed Recording' is synthesis by the time-domain processing of recorded sounds. 'Sampling' constitutes a mere playback of the original sound. There are no implicit control parameters, they must be contrived. Pitch is altered by varying the playback sample rate, and timbre is shaped by applying time-varying filtration. Sustaining notes are simulated by looping the sample. All of these operations introduce perceptible distortions away from the original sound. An alternative is to compile a library of samples that encapsulate the acoustic source's timbral range as a function of its control parameters. However, storing many complete variants of the same basic sound is an inefficient use of memory. Notwithstanding, sampling is a most effective technique where the acoustic source has a minimum of performance control parameters, such as percussion instruments, where a single recording may suffice (Smith, 1991).

'Wavetable' synthesis is a logical development of the looping principle: only one period of the tone is stored eliminating much of the memory redundancy of sampling. Greater storage efficiency facilitates the compilation of a library of wavetables that more

completely characterizes an instrument's timbre range: a recent case is provided by Horner et al (1993). As the timbre of a note is a function of its pitch and evolves with time, an infinite number of tables are required, in theory, to describe this two-dimensional mapping. However, the process is generally smooth and therefore an efficient data-reduction technique is a quantised grid with intermediate timbres generated by linear interpolation between 'nearest neighbour' wavetables (partial phases must be normalized). Quantisation of the pitch and time axes should be as coarse as perceptual constraints will allow in order to minimise redundancy. A suitable example is acoustic piano resynthesis: the similarity of timbre between adjacent notes implies that quantising the pitch axis to individual notes creates redundancy. However, octave-level quantisation, say, is impractical because the fixed-length of wavetable implied at each pitch value has a bandlimited range of operation because (i) they are prone to quantisation noise at lower fundamental frequencies (an oversampled table is a solution) and (ii) higher harmonics may alias at higher fundamental frequencies. Therefore a compromise is often achieved. An advantage of the resulting restricted range of pitch modulation is that the complexity of voice-specific sample rates (c.f. sampling) can be avoided using a phase-accumulator approach at the expense of phase-jitter noise (see section 7.4.2). Mauchly and Charpentier (1987) discuss a typical hardware implementation.

For instruments with a number of interdependent control parameters e.g. a violin with three parameters of pitch, bowing velocity and distance from bridge, linear interpolation becomes multi-dimensional. This is called 'vector synthesis' (Smith, 1991) as the control parameters represent interpolated timbre as a vector in a *n*-dimensional bounded timbre-space (Wessel, 1979). For *n* parameters a minimum of 2^n control points are needed, representing vertices of the bounding hypercube, which can consume a substantial amount of memory space and bandwidth in implementation. It also makes the assumption that with all other parameters fixed, timbre varies linearly between the extreme limits of a single parameter. For timbre as a function of frequency over 7-octaves, this is likely to be unsatisfactory. Non-linearity is supported by PWL subdivision of the hypercube into a lattice (Haken, 1991). However, if the parameters are mutually independent, then only $2n$ wavetables are needed. For instance, FOF synthesis exploits the independence of pitch

and formant envelope in singing voice synthesis with *n* wavetables for *n* formants (Rodet, 1984). Formant envelopes can also be implemented by filter structures (Sandler, 1989).

Wavetables with interpolation represent an attempt to resynthesise the recorded instrument from a small data set. The analysis procedure is one of data-compression rather than the derivation of useful abstractions such as a spectral or physical model. However, the stored waveform approach is flexible, it can create purely synthetic sounds, generated 'off-line'. It is also possible to create impressive and original musical textures with deliberately intensive processing via 'granular' synthesis (Roads, 1978). Of relevance to AS is the fact that some systems have supported its basic principles by manipulating spectra that are written into wavetables via an IFT at run-time (Serra et al, 1990). It is not a full implementation of AS because independent control of partials is lost by their coalescing - for instance, exact harmonicity is assumed. However, the spectral definition is more compact and intuitive than an oversampled wavetable, and makes interpolation easier by default normalisation of partial phase (Smith, 1991).

2.2.2 Physical Modelling

Physical modelling synthesis uses discrete-time mathematical models that describe the physical behaviour of acoustic instruments, which without exception, have a form which includes excitation of a resonant system that possesses control parameters that modify resonant behaviour. Control interfacing mimics the typical interaction of player and instrument. With sophisticated control transducers, real-time physical models become as playable as their acoustic equivalents, giving rise to their classification as 'virtual' instruments. They also encapsulate naturally occurring phenomena such as non-linear and chaotic behaviour which is difficult to map into an inherently linear wavetable model of synthesis: an example is the octave jumps in pitch caused by overblowing on a clarinet. Also, a physical model does not express its timbre space or control mechanism in terms of a static lexicon of stored waveforms, rather it is implicit in the topology of the model, eliminating much potential for redundancy (Smith, 1991).

The most popular building-block for a physical model is the digital waveguide: a delay line that models a wave (pressure or displacement) travelling through a unit length of a one-dimensional medium, such as a taut string. Actually, two delay-lines operating in

opposing directions are needed to simulate bi-directional propagation. The Karplus and Strong (1983) ‘plucked string’ synthesis algorithm was first to indicate the potential of waveguide techniques. The output of a delay-line, initially filled with noise is fed back to the input via a low-pass filter. High frequencies in the initial noise burst decay rapidly leaving a lingering fundamental, determined by the waveguide length. Many other physical models of acoustic instruments dependent on a single resonator have been constructed successfully based on single wave-guides, such as woodwind instruments (Smith, 1987).

For instruments that are composed of interacting resonators, a more complex approach is needed. An example is simulation of the interaction between the forced vibration of a violin string and the passive resonance of the body. ‘Waveguide’ synthesis uses two-dimensional ‘meshes’ constructed with grids of waveguides connected by lossless scattering junctions, which simulate membrane stiffness and provide signal dispersion. Meshes can take the forms of plates and cylinders etc., the topology following the form of system to be simulated (Van Duyne and Smith, 1994). Examples of other constructs in the technology are non-linear reflections at boundaries that shape timbral evolution, and digital hammers that model all of the physical interaction required to simulate natural attack transients (Van Duyne et al, 1994).

The limitation of physical modelling is that one is fundamentally limited to resonant systems and excitation functions, and this is but a subset of musical sounds. Virtual instruments do open new possibilities for performer/instrument interaction but the emphasis on physical control creates a dependence on transducer technology: a conventional MIDI keyboard is an inadequate way to control a woodwind model. Timbre space is more specific than wavetable synthesis which can reproduce arbitrary sounds where the equivalent model would be intractable to compute. Within that space, however, a single model may capture intrinsically what would otherwise require an extrinsic definition through an unrealistically large number of wavetables. As in resynthesis with spectral modelling, the physical approach is extendible to the simulation of systems with physically unrealizable characteristics, for novel but intuitive musical effects.

2.2.3 Abstract Algorithm

Musical tones may be reproduced from recordings, or from modelling the physics of instruments or the evolution of the audio spectrum perceived by the listener. 'Abstract' algorithms do not use natural processes as a reference point in this way. Many numerical algorithms have musical properties by accident, which can be exploited for music synthesis. There chief motive is cost-effectiveness; an abstract algorithm may provide impressive sounds and yet be efficient to map into silicon: an argument that has weakened over the years with increasing VLSI performance. Frequency Modulation (FM) synthesis is the most successful example of this category (Chowning, 1973): an alternative technique is 'discrete summation formulae' (Moorer, 1976). In the simplest form of FM, a carrier sine oscillator at a frequency ω_c is frequency modulated by another oscillator at ω_m with the amount of modulation controlled by the 'index' $I(t)$ as in equation (2.1).

$$y(t) = A(t)\sin(\omega_c t + I(t)\sin\omega_m t) = A(t)\sum_n J_n(I(t))\sin(\omega_c t + n\omega_m t) \qquad (2.1)$$

The spectral decomposition of an FM waveform is a set of sinusoids shaped by a spectral envelope $A(t)J_n(I(t))$ where J_n is the Bessel function of order n. $A(t)$ governs amplitude, the index $I(t)$ governs timbre; at $I(t)=0$, the note is simply the carrier sinusoid, as $I(t)>1$ the envelope takes the form of two peaks either side of ω_c which gradually migrate causing harmonics near ω_c to diminish in amplitude and those further away to increase. Thus, the concentration of harmonic power in the spectrum varies monotonically with $I(t)$: a 'brightness' control that can be enveloped, for instance, to simulate the characteristic "waah" of brass instruments (Higgins, 1990). Another control parameter is the ratio of ω_c to ω_m: an integer ratio gives harmonicity, and a non-integer ratio, inharmonicity (for bell-like tones). The 'folding-over' of negative frequencies creates additional spectral complexity (Moore 1990).

The application of FM principles is scaleable beyond the simple oscillator pair of equation (1.4) With six oscillators per voice (as in the celebrated Yamaha DX-7) the number of possible configurations proliferates, providing a richer sound palette than AS

with the same resources. The key to the computational efficiency of FM is that modifications to the parameters of a single modulating oscillator controls *many* spectral components in a musically useable way, unlike the one-to-one relationship in AS where modifiying a single partial parameter usually has little impact on timbre (it is useful to classify respective parameter behaviour as ‘strong’ and ‘weak’) (Jaffe, 1995). However, musicians are still limited to synthetic, characteristically “FM”-type sounds and parameters that often have counter-intuitive effects compared to those manifest on acoustic instruments: a fact that complicates analysis support: genetic algorithms are proving useful in deriving FM parameters that map on to analysed acoustic timbres (Horner et al, 1992).

In contrast to FM, which uses non-linearity in oscillator phase-accumulation, ‘waveshaping’ creates spectra by applying a non-linear function directly to a sinusoid at the fundamental frequency. Functions for arbitrary (but strictly harmonic) spectra are derived from applying Chebyshev polynomials (De Poli, 1983). Timbre is independent of frequency, but a function of amplitude which evolves towards the specified spectrum at unity amplitude, and distorts beyond. Waveshaping principles overlap with the domain of physical modelling and, indeed, have direct application because non-linearity is a characteristic of the physics of many natural systems. For example, the vibration of a reed is linear at low amplitudes and non-linear at higher (Rodet, 1992).

2.3 Developments in Spectral Modelling

The most successful spectral modelling approach prior to the availability of commercial digital music synthesisers was ‘subtractive synthesis’: a complementary philosophy to AS. A harmonically rich signal, such as a pulse stream is shaped by applying a filter with an appropriate frequency response. Its origins lie in its ease of construction using analogue electronics with voltage-controlled oscillators, amplifiers and filters (VCO, VCA and VCF). ‘Algorithms’ were represented by hardwiring modules to create a ‘patch’ (Smith, 1991). It was a popular technique that became dated for two reasons, (1) simulations of acoustic instruments were judged poor and became superseded by sampling and wavetable technology which derived their sound directly from the acoustic source in question, (2) the spectral envelope of synthetic sound is shaped by low-order

filters and complex envelopes must be built up by adding more modules to a patch. FM synthesis, in contrast, can create novel and complex spectral effects with a simple oscillator topology. The sound, however, remains musically distinctive and a revival of interest in subtractive synthesis is underway as evidenced by (Stilson and Smith, 1996)

Pioneering experiments using AS on digital computers were conducted by Risset (1965) in the language Music V: it was demonstrated that a complete trumpet note could be analysed into a set of partials and satisfactorily re-synthesised using PWL approximations of partial amplitude envelopes (see section 1.1.3). Establishment of the viability of digital AS led to a concentration in providing analysis support to permit the spectral modelling of arbitrary acoustic sources. Due to the limited performance of contemporary mainframes, processing was performed off-line - often overnight - to create recordings that could be played at normal speed (Smith, 1991). Real-time digital synthesis was not yet feasible. In the early days therefore, the computational expense of AS was a matter of inconvenience, and of lesser importance compared to the power of spectral modelling to create any conceivable sound.

The simplest analysis method is the ‘heterodyne filter’: a bank of linearly-spaced bandpass filters implemented by frequency shifts and parallel instantiations of a prototype low-pass filtration. A synonymous technique is the digital ‘phase vocoder’: a sliding Discrete Fourier transform (DFT) which produces phase and magnitude values for every frequency bin each sample period. The DFT is efficiently implemented by the Fast Fourier Transform (FFT). A sinusoid appears as a magnitude peak in one of the bins, and its phase, frequency (from the sum of the inter-frame phase derivative and bin centre frequency) and amplitude parameters may be extracted. Linear bin spacing matches the linear spacing of partials in a harmonic tone. With an FFT of sufficient length, individual partials can be identified by a tracking algorithm which extracts their amplitude, frequency and phase envelopes. Over the years, many improvements have been proposed for the phase vocoder (e.g. optimisation of hop size, window function) and it remains an important analysis / synthesis tool for computer musicians (Moore, 1990).

Availability of analysis support and libraries of analysed tones provided researchers with data to investigate data-reduction methods to make AS more efficient. Early AS engine

designs such as (Snell, 1977) introduced the TOB model of Fig. 1.2 which uses PWL envelope compression. Sasaki and Smith (1980) use a similar design, but propose codifying control information as a pair of two-dimensional arrays for (i) spectral envelopes and (ii) partial frequency ratios where one subscript is a scalar 'control index' which extracts the required instantaneous control data vector for the set of *S* sinusoids. Additionally, there are global parameters of amplitude and frequency. Timbral evolution is effected by applying an envelope to the control indices in a similar way to that used in FM synthesis. A unique feature is the rejection of PWL in favour of low-pass filtration (f_c=100Hz) on envelope streams enabling a low-bandwidth control rate without audible artefacts. Hence, the undefined controllability of AS is resolved by making its control interface 'look' like that of contemporaneous abstract algorithms. However, all control data must be generated *a priori*. Also, generality is necessarily lost by making timbre a function of two control parameters. However, an important concept is introduced into AS for the first time: a single 'strong' parameter or 'metaparameter' is mapped onto a set of 'weak' partial amplitudes (Jaffe, 1995). This is a key strategy for resolving control issues in AS at layers of data abstraction higher than PWL representation.

Given a musical tone expressed as a large set of envelopes in PWL form, another strategy for solving the control overhead is to abstract a higher-level description. Reasons are twofold: further data-reduction is desirable to reduce implementation costs whilst derivation of intuitive 'handles' or metaparameters on the data set improves SME ergonomics. Charbonneau (1981) proposes a data-reduction technique. An amplitude envelope (normalized to 1) of all partials, and the peak value of each partial, is extracted using the product of the two for resynthesis. The same algorithm was applied to frequency functions, and each partial had independent start and finish times between which master envelopes were warped. For short duration tones, satisfactory results are reported, but the scheme does not account for temporal evolution of spectrum over longer durations. In contrast, Strawn (1980) developed a scheme, using pattern recognition techniques, for decomposing an arbitrary PWL function into a hierarchy of diminishing triangles, representing perceptually important 'features' and 'sub-features': it is the abstraction of a metastructure. Redundant information is required to express the hierarchical structure, but modifications can be made at different levels of abstraction i.e.

the overall shape may be changed, or just a single feature. Elsewhere, Strawn (1987) discusses the PWL modelling of transitions between notes in an analysis context.

Schindler (1984) uses a combination of the previous techniques to generate a hierarchy of spectra describing the relative amplitude of features and subfeatures in each partial, with relation to breakpoints on a master envelope. The advantage is that the spectral evolution of a sound is described in a structured form, based on master amplitude and frequency envelopes. This is then integrated with a two-dimensional timbre frame associated with an event e.g. attack as a function of pitch and loudness. The frame is organized into a rectilinear grid with context-specific quantisation of axes. Each point on the grid is associated with a structured envelope set and resynthetic envelopes at an intermediate point are interpolated from the vertices of the bounding rectangle. He then develops methods for timbral evolution within frames, splicing of frames for a chain of events (attack, decay), and proposes a custom architecture. The objective is to use a modifiable, structured representation - after Strawn(1980) - in a data-efficient form (after Charbonneau) with an emphasis on generality.

Kleczkowski (1989) developed Group Additive Synthesis as a data-reduction technique by identifying redundancy within sets of envelope functions. His technique depends upon clustering envelopes into groups, on the basis of Euclidean distance in raw-data form, and extracting a master line-segment amplitude and frequency for each group. Grouping methods are discussed by Eaglestone and Oates (1990) and Horner and Cheung (1995). To discriminate between partials in a group, two vectors of constants describe the relative magnitude of partial amplitude and frequency in relation to the master envelopes. He qualifies his procedure with listening tests to support his hypothesis that only perceptually insignificant data is lost during grouping. As redundancy is eliminated, a more compact control data set is extracted that is simple to uncompress in real-time. Also, hardware savings are possible if a group contains *n* harmonic partials: a single wavetable oscillator initialised by the IFT of the amplitude vector replaces *n* sine oscillators. It is also envisaged that the use of master envelopes makes grouping compatible with the metastructures employed by Strawn (1980) and Schindler (1984).

Another strategy is to prune out weak partials that are psychoacoustically 'masked' by stronger ones in their frequency vicinity. The fundamental idea can be understood in an AS context via a simple conceptual experiment: a listener is presented with two simultaneous tones; (i) a reference sinusoid of constant amplitude and frequency at f_1 and (ii) a sinusoid of variable amplitude and frequency at f_2. The graph of amplitude(f_2) *versus* f_2 defining the threshold where the listener perceives *both* tones, rather than just the reference f_1 (i.e. f_1 is just masking f_2), is the 'critical band' at f_1 which takes the characteristic form of a bell-shaped curve illustrating the bandpass excitation pattern caused by f_1 along the basilar membrane in the cochlea. Below this threshold, f_2 is not perceived as a separate tone and its presence, or otherwise, is perceptually irrelevant. In reality, the interaction (beating) of f_1 and f_2 would be noticed because of the finite time resolution of the ear thus reducing, somewhat, the universal validity of such a simplified method for high-quality AS. Returning to a high-level perspective, the idea of 'receiver coding' is that, given (a) foreknowledge of the critical band properties of the inner ear (via an auditory model) and (b) a partial set described by AS amplitude and frequency vectors, a reliable prediction can be made as to which partials will be masked and thus not require synthesis (Moore, 1990).

An example of the successful exploitation of masking is the MPEG algorithm for high-fidelity audio data-compression at low bit-rates. The bit-rate (and thus quantisation noise) in each sub-band is optimised to shape the full-band noise-floor to the expected masking envelope of the full-band signal spectrum so that the *perceived* signal to noise ratio remains acceptable (Pan, 1994). However, receiver coding of time-varying PWL AS parameters is a different issue. Savings made by minimising the number of active oscillators must be in excess of the pruning overheads for the technique to be economic. An evolving spectrum implies (i) frequent (and expensive) recomputation of the auditory model and (ii) the high probability of intermittent partial masking necessitating frequent oscillator allocation and de-allocation (increasing resource allocation complexity for little re-allocable gain). A simplification is to assume that masking is constant during the duration of a note because of inherent frequency stationarities (c.f. section 1.4.1).

For instance, Marks (1988) allocates static priorities related to partial amplitudes; the higher the value the more significant its contribution is to the perception of timbre. These are used to prioritize resource allocation in a low-complexity AS engine with a value of *S* insufficient to handle the theoretical peak number of partials. Haken (1991) develops this idea further by introducing a simplified auditory model to predict masking effects. The spectrum is decomposed into a series of seven frequency bins approximating critical band spacing. Partials are mapped into bins and are pruned to a fixed number, if such is exceeded, on the basis of retaining those with maximum peak amplitude. In listening tests, it is argued that by exploiting masking effects no more than 75 partials are required for the synthesis of a stationary “symphonic”chord. A real-time implementation uses timbral interpolation within a latticed cubic space (see section 2.2.1). Of interest to this thesis, two sample rates are utilised, with low frequency partials synthesised at the lower sampling rate showing a latent recognition amongst researchers in the application of multirate DSP to AS.

Provided with a group of instruments sounds, it is possible to identify a set of basis functions for additive modelling of the group of smaller size than the equivalent set of sinusoids. Stapleton and Bass (1988) demonstrated that the Karhunen-Loeve (KL) transform can generate such a set to span a group of instrument waveforms that share a strong cross-correlation. Linear combinations of the basis functions reconstruct the individual timbres of the group. Plomp (1976) used similar principles in extracting an optimal set of spectral shapes, in frequency domain. Listening tests confirm the quality of resynthetic tones. The chief advantage reported is the replacement of an unspecified number of sinusoids by a maximum of five basis functions, reducing the amount of real-time computation and control data. Disadvantages include (1) a limitation to harmonic tones and (2), the spectrum defined by a basis function can alias at high frequencies and must be bandlimited to allow frequency variation.

The ‘Sine Circuitu’ (Jansen, 1991) is a typical recent example of a TOB form; a single module can generate 625 sinusoids at f_s=44.1kHz, with linear amplitude and frequency envelope generation under the control of a transputer. Up to sixteen modules can be connected in parallel to provide 10,000 sine generators in real-time. An object-oriented

environment for algorithm development is provided. Such machines are invaluable test-beds for researching the potentials of real-time AS. The chief disadvantage is cost. A number of high speed discrete logic components are required, and the control mechanism is therefore simplified to an operational minimum; discrete integration for enveloping without hardware breakpoint detection or line segment queuing. These low-level tasks are transferred to the controlling transputer software. In contrast, a recent example of a small-scale prototype TOB is described by (Houghton et al, 1995). It is based upon a (standard cell) ASIC, containing control logic and an interpolated LUT (its chief feature of academic interest, see section 7.4.3) with an external multiplier / accumulator and memory, mounted on a plug-in board to an IBM PC which provides real-time control. 127 oscillators are thus supported with envelope generation performed in software.

An early example of the commercial exploitation of digital AS is the BMIS (Bradford Musical Instrument Simulator) developed at the University of Bradford, UK (Comerford, 1993). It is a general purpose architecture comprising up to eight ‘Music Modules’, each offering 64 ‘hardware generators’. Simulation of classical pipe organs is the chief application. The internal architecture of a module differs slightly from the standard pattern by storing waveforms in RAM instead of a sine LUT in ROM. A ‘primary waveform synthesizer’ writes the required waveforms into RAM at run-time from spectral data sent to the module from a host computer via the ‘Musibus’. Savings in computation are achieved in a variety of ways as discussed by Marks (1988) and Kleckowski (1989). Also, a set of partials coincident in frequency from consonant voices in a chord can be replaced by a single sinusoid of equal RMS power to the set.

Spectral Modelling Synthesis (SMS) as developed by Serra and Smith (1990) is currently one of the most popular tools for the music analysis / synthesis. AS models the deterministic part of the signal (composed of a sum of sinusoidal partials) whereas subtractively filtered noise (via an IFFT) models the stochastic part, once the former has been isolated by a partial tracking algorithm. Accurate modelling of sounds which have a strong noise component (e.g. human speech) is thus facilitated. As originally proposed, SMS assumed an oscillator bank model of AS, but its compatibility with AS via the IFFT using the method of Rodet and Depalle (1992) leads to a logical integration of the

stochastic and deterministic components into a single IFFT-AS schema. This proprietary technology is known as Fourier Analysis Resynthesis (FAR) (White, 1995).

2.4 IFFT Simulation of the TOB

The radix-2 IFFT is a widely-used optimised form of the Inverse Discrete Fourier Transform (IDFT) which converts a discrete frequency vector $X(\omega)$ of N bins (where N is an integer power of 2), each describing the magnitude and phase of bin frequency in complex form, into a signal vector $x[m]$ of N samples. For real-time synthesis of a continuous music signal, consecutive IFFT's or 'frames', are required with the Short-Time Fourier Transform (STFT) of the desired signal written into an IFFT each frame where N is chosen to give good time resolution. Use of the term STFT is justified by the observation that N is the length of the FFT / IFFT analysis window (which is by default rectangular). The significance for AS is that a controllable sinusoid may be synthesised by predicting its STFT each frame after which N samples are automatically generated by the IFFT, in contrast to the computation required *each* sample period in a TOB. By superposition, a single IFFT can accommodate any number of sinusoids with the limitation that they are accumulated into a single sample stream. For a suitable reference of IFFT musical applications see Chamberlin (1980).

The control data rate is reduced from the sample rate to the frame rate: Rodet and Depalle (1992) propose a typical value of 128:1. Step changes in parameter values between frames generate high frequency interference that is distracting for the listener. To counteract this, smooth amplitude and frequency interpolation is required between frames to make the coarse resolution of control imperceptible. Also, control resolution has a lower bound due to Heisenberg's principle (as discussed in section 1.4.2). For instance, the synthesis of fast rise-time transients is difficult if N is too large. Hence the frame size should be a compromise between small N for sufficient control resolution and large N so that the condition in eqn. (2.2) is satisfied where c_{TOB}, c_{IFFT} and c_{STFT} are, respectively, the costs of a TOB oscillator update, IFFT computation and STFT computation for a single sinusoid (Freed et al, 1993).

$$\frac{Sc_{STFT} + c_{IFFT}}{N} < Sc_{TOB} \tag{2.2}$$

2.4.1 Overlap-Add Synthesis

Early proposals for IFFT-AS synthesis mapped one sinusoid to one IFFT bin providing only $N/2$ harmonics of the frame frequency for synthesis (Chamberlin, 1980). Exact frequency is maintained by rounding the desired frequency to the nearest bin and introducing a phase lead or lag each frame to approximate the residual frequency fraction. With non-overlapping rectangular windows, phase discontinuities appear at the frame interfaces. However, Overlap-Add (OLA) synthesis with raised cosine windows smoothes out the discontinuities and performs the desired interpolation between the different amplitude values that a sinusoid may take (according to $A_i[n]$) in each frame. In OLA the sum of overlapping windows is always unity as illustrated in Fig. 2.1 with triangular windows. However, twice the number of IFFT's are required. The resulting signal suffers from amplitude modulation because of smeared artefacts of phase discontinuities (Chamberlin, 1980). Smearing effects are avoided in the 'FFT^{-1}' algorithm of Rodet and Depalle (1992) to which the rest of this section is devoted: for other pertinent OLA analysis-synthesis proposals see George and Smith (1992), McAulay and Quatieri (1986, 1987, 1988) and Fitz et al (1992).

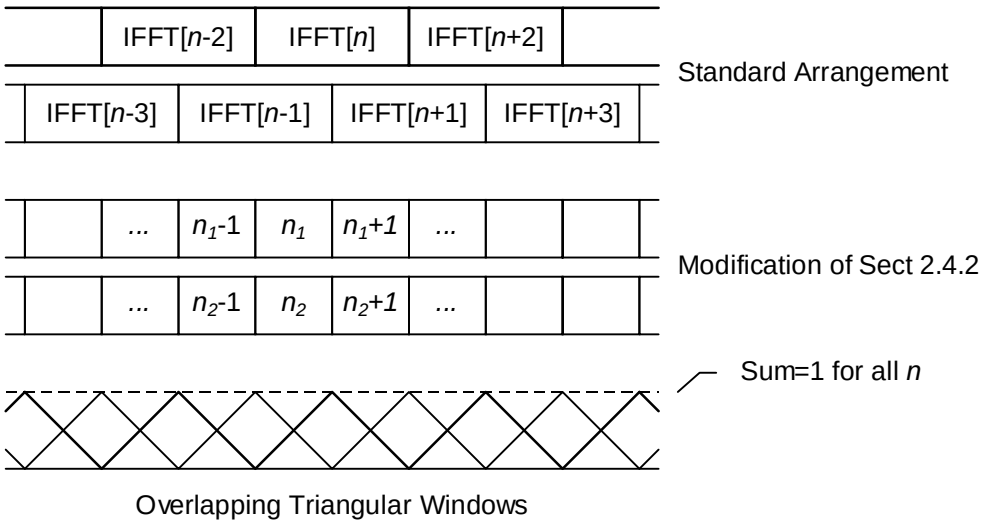

Figure 2.1 Overlap-Add IFFT Synthesis with Triangular Windows

According to the modulation theorem, a stationary sinusoid at frequency F, multiplied by a window $w[m]$, is the frequency domain convolution of their Fourier transforms; the delta function $\delta[\omega-F]$ (for a cosine) and $W(\omega)$. In other words, a displacement of $W(\omega)$ to F. By suitable choice of $W(\omega)$, most of the energy is concentrated near the maximum which is significant as only a small number of look-ups in an LUT approximation of $W(\omega)$ within a narrow margin of the main lobe are necessary to approximate the STFT of a sinusoid for arbitrary F: Brown and Puckette (1992) use a similar strategy in an analysis context. According to Rodet and Depalle (1992), FFT^{-1} uses an oversampled $W(\omega)$ table of length 512 requiring 9 STFT look-ups, and an IFFT of length N=256. The STFT of the resultant unity-gain cosine at frequency F is scaled to the desired amplitude A and rotated about the frequency axis so that its phase matches the previous frame which is achieved by accumulating the phase difference between frames, determined as $N\pi F/f_s$ for time-invariant F. Finally, the STFT is accumulated with those of other sinusoids prior to IFFT / OLA synthesis.

A triangular window $t[m]$ for OLA synthesis, illustrated in Fig. 2.1, performs linear amplitude interpolation between frames and is more desirable than a raised cosine as suggested by Chamberlin (1980) which, for instance, connects frame-rate quantised breakpoints by scaled versions of the curve $\sin(x)$:$-\pi/2 \le x \le \pi/2$, giving a ripple effect for a sinusoid increasing linearly in amplitude across frames. However, the Fourier transform of $t[m]$, $T(\omega)$, has significant smearing of frequency and therefore poor properties for $W(\omega)$ and so an alternative window (which has good spectral resolution) is chosen to construct the STFT of each sinusoid (e.g. Hamming): Nuttal (1981) provides a tutorial on suitable analysis windows. Finally, to ensure that the IFFT of the STFT - windowed by $w[m]$ due to its construction by $W(\omega)$ - generates a signal windowed by $t[m]$ for OLA, the output of the IFFT is post-multiplied by $t[m]/w[m]$.

2.4.2 Supporting Frequency Envelopes

The inclusion of frequency enveloping in the FFT^{-1} algorithm requires a mechanism that performs frequency interpolation between consecutive frames (from $F_x[n-1]$ to $F_x[n]$) for a sinusoid x. A solution proposed by Goodwin and Rodet (1994) is to synthesise chirps within frames; nonstationary sinusoids changing linearly in frequency from $F_x[n-1]$ to

$F_x[n+1]$: such end-points are determined by the OLA algorithm. The STFT of a chirp is no longer a simple displacement of $W(\omega)$, but a broader version with a breadth dependent on $\Delta_F = F_x[n+1]-F_x[n-1]$ and centred upon $(F_x[n+1]+F_x[n-1])/2$. As the distortion of $W(\omega)$ is dependent only on Δ_F, and $W(\omega)$ is a compact LUT, a two-dimensional array $W(\omega,\Delta_F)$ is a convenient representation: a coarse resolution (in terms of Δ_F) with linear interpolation is favoured. Once $W(\omega,\Delta_F)$ is derived and displaced to $(F_x[n+1]+F_x[n-1])/2$, the processing steps are identical to the case of a stationary sinusoid. Maintaining correct inter-frame phase, however, requires a quadratic formula to take into account the parabolic phase trajectory of the chirp.

The proposed modification assumes that Δ_F is constant in consecutive frames. If Δ_F changes between frames, then a splicing error occurs because of an irreconcilable phase mismatch between the different chirp rates in the overlap region: discontinuity side-effects of IFFT / OLA synthesis re-emerge in the first-derivative of $F_x[n]$. A further refinement proposed by Goodwin and Kogon (1995) that transposes the error to the second-derivative is to divide frames in two and match the chirp rate in the first half of a successor frame with that in the second half of its predecessor i.e. $\Delta_F = F_x[n]-F_x[n-1]$. For linear amplitude interpolation between frames, a chirp of amplitude $A_x[n-1]$, multiplied by the falling half of $t[n]$, is summed with a chirp of amplitude $A_x[n]$, multiplied by the rising half of $t[n]$. Two IFFT's of length $N/2$ are thus required to permit the superposition of independently controllable sinusoids as illustrated in Fig. 2.1. The same STFT is used for both chirps of x, scaled to $A_x[n-1]$ and $A_x[n]$ for the IFFT pair.

2.4.3 Noise Synthesis

Noise synthesis is straightforward with IFFT / OLA synthesis. As in SMS (Serra and Smith, 1990), the magnitude spectrum of the desired noise is written into an IFFT with random phase each frame to avoid periodicity. SMS models the residual noise envelope as a PWL spectral envelope each frame with equispaced breakpoints in frequency. This is permissible as the noise envelope is smooth and peaks, representing sinusoids, are extracted beforehand by a partial tracking algorithm. Residual noise representation can be minimised further by quantising the spectrum into Equivalent Rectangular Bandwidths (ERB's) via an auditory model and computing the noise power required in each ERB

(Goodwin, 1996). By superposition, both noise and sinusoidal synthesis may be integrated in the same IFFT.

2.4.4 Performance Evaluation

FFT^{-1} is primarily a software algorithm due to the complexity of operations required to generate the STFT of nonstationary sinusoid. For maximum performance, attention is paid to the instruction and data tables sizes to ensure that critical sections of the algorithm are accommodated in the CPU caches and accesses to slow external memory are minimised. According to Freed et al (1993) implementation of FFT^{-1} in 'C' on a MIPS R4000 workstation (internal clock 100Mhz) shows that a maximum of 310 oscillators are supported at f_s=44.1kHz whereas a software oscillator bank yields 55: a speedup of approximately 6. Freed et al (1993) reasons about the inefficiency of VLSI solutions and discusses a hypothetical 50Mhz monolithic VLSI oscillator bank requiring 4 clock cycles per update. This is calculated to yield a maximum of 290 sinusoids. The conclusion is drawn that cost-effective AS resources are most efficiently provided by a combination of efficient coding of FFT^{-1} (via optimising compilers) and high-clock rate CPU's with sufficient caching to support the code and data.

The STFT operations in FFT^{-1} represent a pre-processing kernel which takes the standard PWL AS envelopes of $F_i[n]$ and $A_i[n]$ and transforms them into a form suitable for IFFT / OLA synthesis. The interface is cast to resemble a TOB except that breakpoints are quantised to the frame rate of $2f_s/N$ rather than f_s. The optimality of FFT^{-1} has an upper bound because increasing N implies a finer resolution IFFT which, in return, requires a finer resolution of $W(\omega)$ in the STFT to maintain synthesis quality. However, T_{max} from section 1.2.2. places an upper bound on N because of delays, proportional to N, in blockwise IFFT processing (Jaffe, 1995). However, the requirement for sufficient control resolution as discussed in section 2.4 - which is finer than that of T_{max} - means that latency is not a significant problem with FFT^{-1}.

One method for representing control data is to superimpose PWL enveloping over a coarser time grid. For line-segments longer than a frame, control data for intermediate frames is interpolated at frame-level. However, PWL presumes that the next breakpoint is known *a priori*, precluding gestural control by metaparameters (Jaffe, 1995). In

contrast, the technique suggested for FFT^{-1} is to store a file of spectral envelopes indexed by a metaparameter (Sasaki and Smith, 1980) for compatibility with analysis tools such as SMS which convert a musical signal into a sequential file of spectral frames describing the noise spectrum and a set of oscillator frequency and amplitudes. If the analysis file is indexed sequentially, then exact resynthesis occurs. To effect certain key time-domain transformations, however, linear interpolation between frames *is* required. One example is time-stretching without altering pitch. Hence, IFFT-AS has the net effect of creating a coarser PWL time grid than that presumed by the TOB. A single low control rate of f_s/N is imposed in place of a higher, redundant rate of f_s.

2.5 Review

In outlining the literature relating to AS, it emerges that though the definition of AS is succinct - eqn. (1.1) - the question of control remains open-ended. Researchers have addressed this control problem from two different angles; (i) spectral modelling paradigms for supporting analysis parameters in a data-efficient, modifiable and intuitive manner and (ii) optimising oscillator bank design, with an intermediate representation between these layers of PWL envelopes. This situation displays little evidence of change, and major future innovations lie in the area of making implementations more powerful, useable and affordable. A variety of strategies have been proposed, some of which compromise generality - a fundamental advantage of AS - for apparent performance gain. From the evidence presented, the conclusion is drawn that the long-term application of an AS data-reduction technique is related to how generally it can be applied.