1. Introduction

1.1 A Tutorial on 'Classical' Additive Synthesis

1.1.1 Definition and Properties

$$y[n] = \sum_{i=1}^{S} A_i[n]\sin(\Phi_i[n]) \text{ with } \Phi_i[n] = \Phi_i[n-1] + \frac{2\pi}{f_s} F_i[n] + \phi_i[n] - \phi_i[n-1] \qquad (1.1)$$

$$\begin{aligned} A_i[n],\ F_i[n],\ \phi_i[n] \ &= \ \text{amplitude, frequency and phase envelopes of } i^{\text{th}} \text{ sinusoid} \\ f_s &= \text{industry-standard digital audio sample rate e.g. CD at 44.1kHz} \\ n &= \text{sample index} \end{aligned}$$

The digital form of Additive Synthesis (AS) as defined in equation (1.1) is a formulation of the Inverse Fourier Transform (IFT) where the superposition of S discrete-time sinusoidal oscillators in time-domain maps to a spectrum of S discrete lines in frequency domain (De Poli, 1983). Given a sufficiently high value of S and independent control over each oscillator (in terms of frequency, amplitude and phase envelopes), signals with a rich time-evolving spectrum can be synthesised. A particular application is the synthesis of musical tones which can often be expressed as a sum of harmonically related sinusoids or "partials". Each partial is mapped to an oscillator and the spectrum of the tone is quantified by two vectors of length S (for frequency and amplitude). These determine the 'spectral envelope' of the signal and give precise control over the quality of timbre perceived by the listener: a phenomenon first observed by Helmholtz (1863). The application of AS to music can be summarised by four characteristic properties:-

- Simplicity. AS is completely defined by the 'one-line' eqn. (1.1) and is therefore compatible with a VLSI implementation philosophy (Houghton et al, 1995).
- Analysis support via the Fourier transform to enable resynthesis of acoustic sources. Realistic reproduction of familiar sounds is an indispensable feature of a successful music synthesis algorithm (Risset and Mathews, 1969).

- Generality; it can model the spectral evolution of arbitrary sounds, unifying resynthesis of acoustic sources and “conceptual” sound generated from artificial control data. Traditionally, these domains have been logically distinct e.g. sampling and FM synthesis (Smith, 1991).
- Transformations on spectral data have an intuitive relationship with the listener’s perception. Common operations are more logical to express in frequency than time domain (e.g. pitch shifting, time stretching). Combined with the algorithm’s generality, expressive customisable control interfaces are facilitated (White, 1995).

1.1.2 Disadvantages

AS has a high computational cost in its native form of eqn. (1.1) because of (i) the overheads of execution itself and (ii), more critically, the unresolved problem of how control information is generated in the first place (Rodet and Depalle, 1992). Two streams of envelope data ($A_i[n]$ and $F_i[n]$) are incident upon each oscillator, each of which forms but a small additive energy component in the final audio signal $y[n]$: dynamic phase enveloping via $\phi_i[n]$ is usually omitted if inter-partial phases are correctly initialised and harmonicity is assumed (Horner and Cheung, 1995). A useful analogy for AS is that of data-compression where the incident control bandwidth far exceeds that of the output. A contrast is drawn with classical parametric synthesis techniques (e.g. FM synthesis - section 2.2.3) which have the opposite form. High control bandwidth is attributable to the combination of the three following factors:

- Envelopes should be updated, in ideal circumstances, at f_s to permit the accurate synthesis of transient detail and avoid the appearance of modulation sidebands in the spectrum caused by coarse time quantisation in envelopes (Jansen, 1991).
- An instrument voice will typically require >20 partials for satisfactory resynthesis (many higher harmonics lie below the perceptual threshold and can be omitted) (Moorer, 1977). The lower the note, the more are required: 80 partials is not an untypical requirement for bass notes with a significant treble component. (Haken, 1991)

- To synthesise an ensemble, one should ideally superimpose an instantiation of each sounding note: in practice, the slight detuning of actual instruments generates a rich texture (Meyer, 1993). In contrast, sampling is more economic but offers little opportunity for expressive control (see section 2.2.1.) (Smith, 1991).

Hence a 100-note ensemble with, say, 40 oscillators per note will require a net control rate of $100 \times 40 \times 2 \times f_s = 352.8\times10^6$ samples per second assuming f_s=44.1kHz. Using 16-bit envelope samples, this bandwidth represents about one CD's worth of data per second (600MBytes). Clearly, this is outside the realm of low-cost microcomputer implementation in the present-day which is our desired goal. A more typical supercomputer-based strategy is exemplified by Kaper et al (1995).

1.1.3 Piecewise Linear Envelope (PWL) Modelling of Envelopes

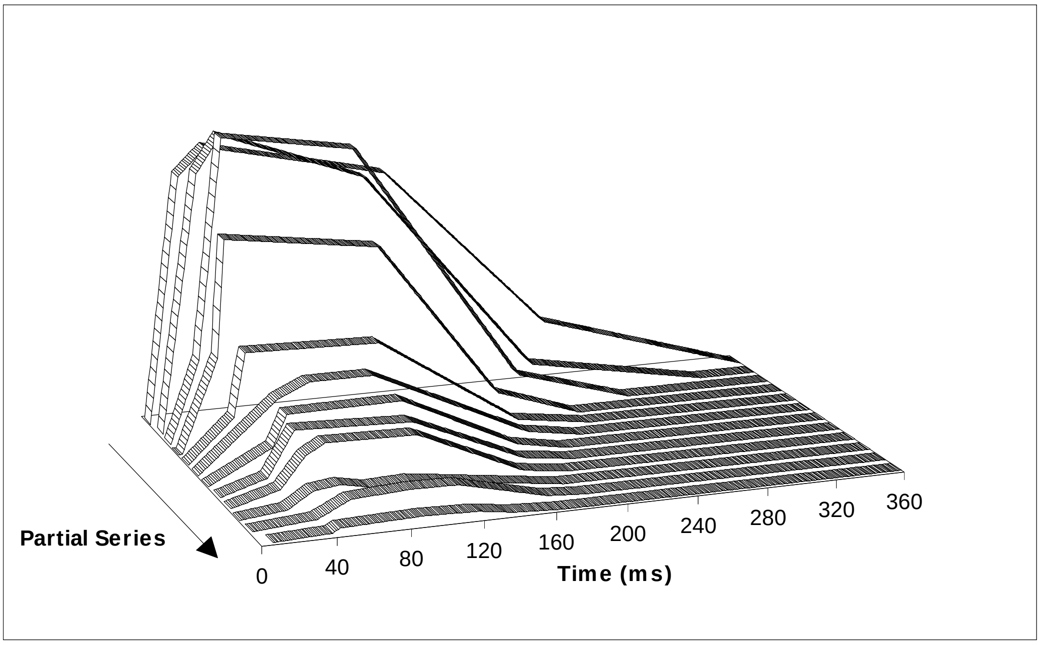

Figure 1.1 PWL Approximation of a Trumpet Note

A significant reduction in control bandwidth is accomplished by using piecewise linear (PWL) models of raw envelope data extracted from a Fourier analysis of acoustic sources (see section 2.3). Such envelopes are characterised by jitter superimposed upon a smooth profile. The latter is preserved as a compact set of breakpoints coded as (time, height) by PWL modelling once jitter is smoothed out (Moore, 1990). Grey (1977) demonstrated via listening tests that resynthesis from a PWL model is musically indistinguishable from the original indicating that envelope jitter is perceptually superfluous. Fig. 1.1 illustrates the PWL amplitude envelope set for the first twelve partials of a complete trumpet sample; an often-cited example of Grey's work which uses 176 breakpoints to model $A_i[n]$ and $F_i[n]$ for $1 \leq i \leq 12$ (Moorer et al, 1978). The data set is thus 352 samples as compared to 360ms $\times f_s$ (44.1kHz) = 15876 samples, or about 45:1 compression. Practical compression factors of 100:1 are reported elsewhere by Serra and Smith (1990). To extend the analogy of section 1.1.2, a static PWL frequency-domain representation of breakpoints is uncompressed into a set of envelope time-series, representing the time-evolving spectrum, and re-compressed by AS into a time-domain image.

The technique is flexible in that, if desired, high-resolution detail such as high frequency transients may be encoded with greater precision and a corresponding penalty in efficiency. Transformation of compressed envelopes is computationally efficient because only breakpoints require manipulation, and intuitive because the 'topographical' representation of a time-evolving spectrum is retained. Uncompression in real-time is simplified if breakpoints are recoded as (*gradient*, *height*) as only discrete integration and a breakpoint detection mechanism are necessary for envelope synthesis (Snell, 1977). PWL modelling is a milestone in the development of AS as a practical tool because it reduces the control data problem from an intractable level to a highly manageable one, without a significant diminution of the properties set out in section 1.1.1.

1.2 On the Specification of an Ideal Oscillator Bank

The properties required of a successful AS implementation may be divided usefully into (i) performance criteria which should be satisfied as far as possible and (ii) concrete design constraints which must be satisfied.

1.2.1 Performance Criteria

- The unit cost of an oscillator should be as low as possible (calculated by cost of oscillator bank / S) to facilitate an affordable additive synthesiser with high S.

- A sufficient number of output streams should be available for spatialisation in the construction of the final multichannel sound image.
- Functional Transparency. Implementation details should not seriously compromise the advantages of AS as set out in section 1.1.1.

1.2.2 Design Constraints

- The oscillator bank output satisfies an industry-standard digital audio format, e.g. 16-bits at f_s=44.1kHz (CD), to ensure high fidelity synthesis.
- Support for linear interpolation between the breakpoints of PWL envelopes for $A_i[n]$ and $F_i[n]$ at a resolution of f_s for high fidelity PWL-AS.
- Real-time system latency from the reception of an event (e.g. MIDI note on) to signal output should be no greater than $T_{max} \cong 10^{-2}$s for imperceptibility.

1.3 Conventional Solutions to AS Computation

1.3.1 The Analogue Background to Digital AS

To understand the importance of AS to music today, it is useful to trace briefly its ancestry. Its origins lie in Pythagoras' theory, derived from experiments with string lengths in harps, that harmony and consonance in music are related to frequency ratios of the numbers 1,2,3 & 4 which is explained physically by the interaction of harmonically related partials (Taylor, 1965). Then, in 1822 whilst working on the physics of heat flow, Fourier developed his theorem and laid the foundations for mathematical signal analysis. Arbitrary periodic functions could be explained in terms of a 'Fourier' series of harmonically related sinusoids at individual phases and amplitudes (Oppenheim, 1983). These ideas were taken up by Helmholtz (1863) as they corresponded with his practical investigations into the physics and perception of sound. He was able to demonstrate that the quality of musical tones was dependent on the spectral envelope of its partial series. Helmholtz also built one of the first additive synthesisers using an electrically driven tuning-fork for each partial with an adjustable acoustic resonator to regulate amplitude.

The telephone, invented by Bell in 1876, represented sound as a varying electrical current and enabled the construction of electromechanical additive synthesisers. Early prototypes involved substantial engineering. Cahill's telharmonium used gearing to establish harmonic ratios and alternators to generate waveforms which were mixed by a resistor network and transmitted direct to earpieces at subscribers' households. Shafts up to 30 feet in length were necessary to carry alternators of sufficient power (Manning, 1985). Subsequently, invention of the thermionic valve and loudspeaker permitted amplification of audio signals and consequently electromechanical tone generators could be miniaturised and manufactured cheaply enough to be put into commercial products. Two famous examples are the Hammond (electromagnetic) and the Compton Electrone pipeless organs (electrostatic), both creating complex waveforms by controlled mixing of harmonically related 'primitive' waveforms (Comerford, 1993).

The introduction of semiconductor and digital computer technology (late 1940's) made many new synthesis techniques possible. AS using rotating tone wheels was superseded first by analogue (1960's/70's) and then digital waveform generation (1980's). However, the concept was adopted by computer music researchers because it is a mathematically rigorous and general way to theorize about music though expensive to compute. As a result of the research effort of the last thirty years, a better perspective has emerged. AS is seen as the foundation layer of a wider 'spectral modelling' paradigm that allows composers to work intuitively with sounds in frequency domain (Smith, 1991). The development of digital AS, as opposed to analogue, is documented in section 2.3: 'AS' and 'digital AS' are interchangeable terms nowadays.

1.3.2 The Traditional Oscillator Bank

A 'direct-form' implementation of AS differs little from the electromechanical schemes outlined. All that is needed is a bank of *S* identical, independently controllable sine oscillators. More usually, given the higher relative clock rates of digital integrated circuits compared to sample rates for digital audio ($>10^7$Hz versus $<10^4$Hz), a single sine oscillator multiplexed *S* times suffices. It is expressed in the previous 'c' code loop which includes the discrete integration required for PWL envelope uncompression (omitting breakpoint detection for clarity). *S* iterations occur every sample period, accumulated to form a single sample op.

```
op=0;
for (n=0;n<S;n++)    {
     amp_acc[n]+=amp_inc[n];
     freq_acc[n]+=freq_inc[n];
     phase_acc[n]+=freq_acc[n];
     if (phase_acc[n]>PI)
     phase_acc[n]-=2*PI;
     op+=am_acc[n]*sin(phase_acc[n]);
}
```

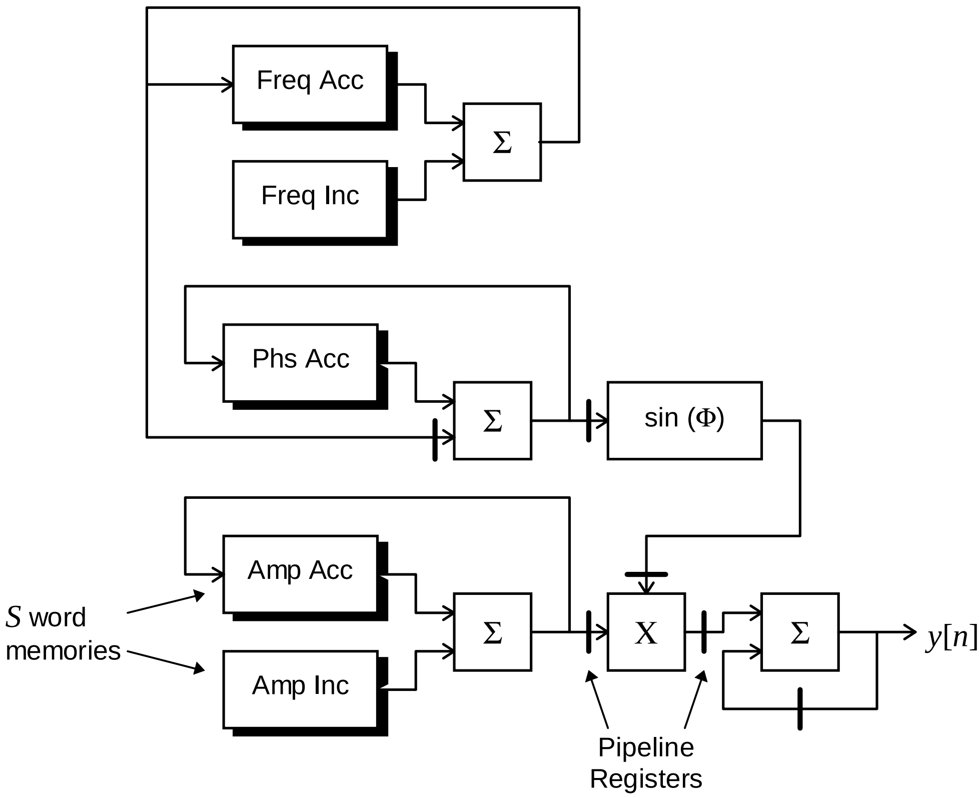

Figure 1.2 Traditional Oscillator Bank

In this form, the mapping to a dedicated architecture, as illustrated schematically in Fig. 1.2, is trivial. The data dependencies between functional blocks are first order and hence pipeline registers may be included as indicated; a *systolic* representation of dataflow within the AS algorithm (Leiss, 1995). The depth of logic and hence, propagation delay, between registers is minimised so that the clock speed, and value of *S* can be maximised. The chief bottleneck is the computation of sin($\Phi_i[n]$) which is a non-linear transformation and often performed by a Look-up Table (LUT). Concurrency in both computation and state memory access is fully exploited. In contrast, execution of the code loop on a standard scalar microprocessor would result in a memory bottleneck for state variable access (Freed et al, 1993).

Within the terminology of parallel computing, AS is a *fine-grain data-parallel algorithm*. Fine-granularity is due to the low process complexity; one iteration of the code loop, in contrast to a large process set size, *S*. It is data-parallel because of the incidence of many independent streams of control information (in the PWL form) upon parallel instantiations of the same process. With a strongly classifiable algorithm, the most effective implementation is often a dedicated architecture. It is a philosophy of "form follows function". Positive evidence is the fact that the systolic model has survived several generations of research prototypes as exemplified by Snell (1977), Jansen (1991) and Houghton et al (1995). A convenient label for this systolic architectural form is the Traditional Oscillator Bank (TOB).

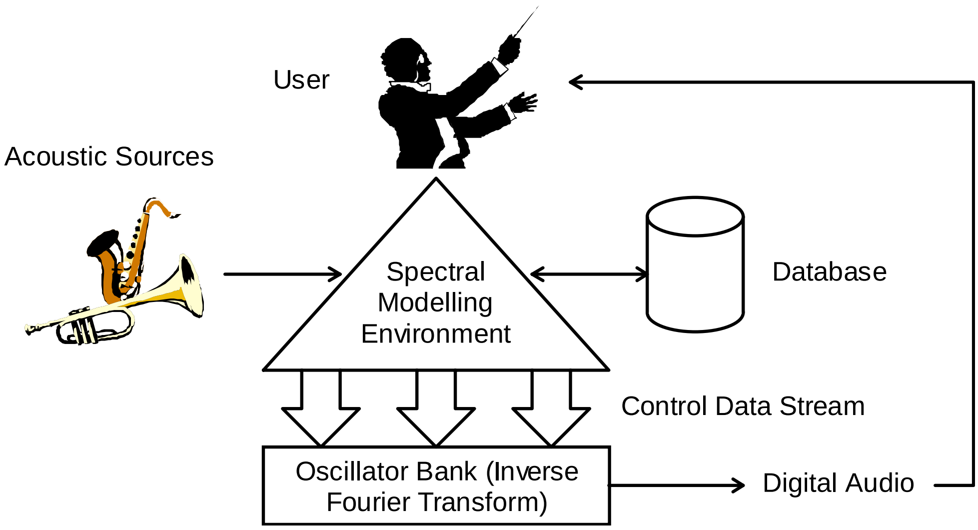

Figure 1.3 The Role of an Oscillator Bank in Spectral Modelling

To adopt a contrasting 'top-down' perspective consider Fig. 1.3, an overview of the complete Spectral Modelling Environment (SME). A recent example is 'Lemur' (Fitz and Haken, 1995). The reader is also referred to Osaka (1995), Freed (1995), Hill (1991) and Feiten and Ungvary (1990). Two distinct systems can be identified; (i) modelling and (ii) synthesis. Modelling uses high levels of data abstraction and is a natural application for software. Powerful CPU’s permit sophisticated object-oriented, graphical SME’s. Towards the base of this pyramidal hierarchy, parallel streams of control data are generated in real-time. An oscillator bank is necessary to perform the IFT represented by AS and should have sufficient synthesis resources to underpin the higher levels of the hierarchy. In contrast to modelling, an oscillator bank is a deterministic process and a natural application of custom hardware, because of the absence of an instruction stream. Reiterating the point of section 1.1.2, functionality is characterised by data *transformation* rather than data processing.

1.3.3 Impact of New Technology

The issues of section 1.3.2. must be set in the context of reductions in VLSI device geometry, permitting smaller faster devices. As a result, both clock rates and packing density have risen giving a net CPU performance increase of about 40% per annum, sustained for the last twenty years. Many synthesis algorithms can now be implemented in software in parallel in real-time on a single high-performance CPU, including forms of AS (Smith, 1991). The expense and lead-time of developing dedicated hardware is no longer understood to be necessary because of innovations in ‘software’ synthesis algorithms which can be ported to newer faster platforms, provided that they share a common operating system such as UNIX. Smith (1991) justifies this bias towards ‘software’ synthesis.

> *“Another problem with supporting special-purpose, computer-music hardware is that it can be obsolete by the time its controlling software is considered useable”*

As explained in Chapter 2, physical and spectral modelling are, in conjunction, two of the most active areas within music technology research in the 1990’s because technological advance is making the implementation of their computationally intensive algorithms (including AS) cheaper and thus available to a wider public. The conceptual framework of modelling provides the necessary “degrees of freedom” required by creative artists. In contrast, algorithm-oriented approaches (see section 2.2.3), typical of the first generation of digital synthesisers in the 1980’s, are falling into increasing disfavour: at that time, economy of implementation was the primary objective and resulted in restricted sound palettes with idiosyncratic control mechanisms. At present, spectral modelling lags its physical counterpart in commercial exploitation due to the problem of high control bandwidth - but its long-term future is secure for the reasons set out in section 1.1.1. Smith (1991) makes an unambiguous prediction:-

> *"It is anticipated that synthesis in the future will be dominated by spectral and physical modelling."... "Spectral modelling is the more general of the two..."*

There is, therefore, both a motive and means to develop products using AS because the commercial potential is large. In deference to the philosophy discussed previously, researchers are looking at AS from a software synthesis perspective. For instance, an alternative to an oscillator bank is the Inverse Fast Fourier Transform (IFFT), the most efficient way to transform a spectrum into a waveform (as discussed in section 2.4). It is a block transform and an outer kernel of pre/post-processing is necessary for effective simulation of the sample-level execution of equation (1.1). Its complexity leads more naturally to a software rather than hardware implementation, but executing on a general-purpose CPU it offers performance competitive to a TOB and is much cheaper to build: partly because 'flagship' CPUs developed by major companies run at the highest clock rates of all (Freed et al, 1993). Therefore, interest has declined in refining dedicated synthesis architectures. Serra (1994) summarises this attitude:

> *"An additive synthesis implementation based on the IFFT has been proposed which is more efficient than the traditional oscillator bank..."*

1.4 Thesis Research Direction

1.4.1 Optimisation of Sample Rate in Note-Based AS

$$c_{as} = aSf_s \tag{1.2}$$

The substance of this thesis is to investigate how far criticism, as expressed in section 1.3.3, about TOB inefficiency is justified given the strength of the arguments in its favour as set out in section 1.3.2. An intial step is to observe that the cost c_{as} of computing AS via equation (1.1) is given by equation (1.2) where *a* is the cost of a single oscillator update. PWL modelling is an economic SME representation but, for conversion into a time-domain signal for playback, uncompression is required in real time and hence benefits of the technique fall outside the scope of eqn. (1.2). Control data bandwidths of the order discussed in section 1.1.2 recur and the implications of equation (1.2) cannot be avoided, unless a software IFFT solution is used. To reduce c_{as} there are three avenues of promise;

1. Reducing *a* by low-level optimisations to oscillator bank hardware
2. Reducing *S* by selective removal of partials from the synthesis set
3. Reducing the (mean) oscillator sample rate f_s.

This thesis concentrates upon developing an algorithm to reduce f_s and an architecture to reduce *a*, leaving *S* as unconstrained as possible so that the subjective qualities (e.g. sound richness) of AS are unimpaired. Optimisation of *a* is mature after a protracted evolution of TOB prototypes, but it is asserted in Chapters 8 and 9 that higher level refinements in synchronisation and scheduling are still possible. Minimising *S* is already practiced in that only the highest energy partials are necessary for acceptable resynthesis quality: 'receiver coding' is a more sophisticated (and problematic) technique which is discussed in section 1.3. The reason for setting f_s to e.g. 44.1kHz is that the audio spectrum extends up to >20kHz and reducing f_s to e.g. 8kHz would produce 'telephone-quality' music of unacceptable fidelity. Hence this strategy is not an option.

Fortunately, this thesis proposes a more subtle approach for reducing f_s which is predicated on the universal acceptance of note as a fundamental concept in music. Ironically, computer-generated music is free from the physical laws that give rise to notes in acoustic instruments, and the universality of note is less valid for electroacoustic composers. However, the unifying factor in all music is that it is perceived by the human auditory system which has a well-defined anatomy and cognitive behaviour, studied through psychoacoustics. Holst (1963) elucidates that note-based music is founded on structures present in spoken language and that there is a phonetic, syntactic and semantic correlation between the two. Notes represent the lowest, phonetic level of musical language and have certain, definable properties, just as utterances in speech have a symbolic notation drawn from a small set of phonemes which are determined by vocal tract anatomy.

The TOB is generalised for arbitrary digital audio frequency signals, rather than just note-based music. All oscillators have a default frequency ceiling, according to Nyquist's sampling theorem, of $f_s/2$ implying that, during their lifecycle, they require frequency dynamics over the entire audio range. This is an erroneous assumption for note-based music, explaining the discrepancy between the computational expense of AS on a TOB and its apparent efficiency of implementation as highlighted in section 1.3.3. In AS terms, notes are characterised by (1) a finite lifecycle of attack, decay, sustain and release and (2) an evolving pitched timbre definable as partial series with $F_i[n]$ and $A_i[n]$ envelopes. Superimposed upon this is "expression" in the form of dynamics, such as tremolo, vibrato or glissando. However, the concept of note remains an intrinsic and absolute determinant in musical discourse.

1.4.2 Determination of Optimal Sample Rates

The observation is made that at or before note onset, data is available on pitch, expected pitch modulation range, and the relationship of partial frequency ratios to modulated pitch (usually considered as constant) which enables the upper bound of $F_x[n]$ for each partial x to be estimated as $f_{max}(x)=\max(F_x[n])$. This data is an attribute of the voice to be synthesised and is therefore usually available *a priori* by default. Given that AS uses sinusoidal basis functions, Nyquist's sampling theorem may be re-interpreted to state that the optimum, or *critical*, time-invariant sample rate for alias-free synthesis of x is given by eqn. (1.3) where, generally, $f_{opt}(x) \ll f_s$. Not only is an oscillator's update rate reduced, but so also is the bandwidth of PWL envelope uncompression. Potential savings are indicated by the ratio $f_s / f_{opt}(x)$ and the repercussions are far-reaching.

$$f_{opt}(x) = 2 f_{max}(x) \qquad (1.3)$$

For illustration, Fig. 1.4. plots the $F_i[n]$ envelope set ($1 \le i \le 12$) for Grey's trumpet as a counterpart to Fig. 1.1 (Moore, 1990). A rapid rise in partial frequencies is evident during note attack which stabilise at harmonic intervals during sustain, but at no point do trajectories stray above 3.6kHz, which implies a maximum $f_{opt}(x)$=7.2kHz which is about 6:1 as efficient as synthesis at f_s=44.1kHz. Indeed, for lower frequencies such as the fundamental at 300Hz, $f_{opt}(x)$=600Hz which implies an efficiency ratio of about 74:1 on the basis of the previous example. A note has a theoretical series of harmonics extending up to ≅20kHz. For instance, with f_x=27.5Hz (A0), 727 sinusoids are required posing a formidable computational burden. However, the logarithmic frequency response of the human ear and the general observation (Sandell, 1994) that the spectral envelope of stationary musical timbres converges towards zero amplitude above 5kHz (disregarding some notable exceptions like a muted trumpet) both indicate that low frequencies should have greater priority for inclusion in the set of synthesised partials than high frequencies in order to reflect their relative perceptual weighting (‘receiver coding’ is elaborated upon in sections 1.3 and Chapter 6). Critical sampling therefore facilitates large savings in AS throughput because computation of these low frequency partials at the non-critical rate of f_s is highly inefficient.

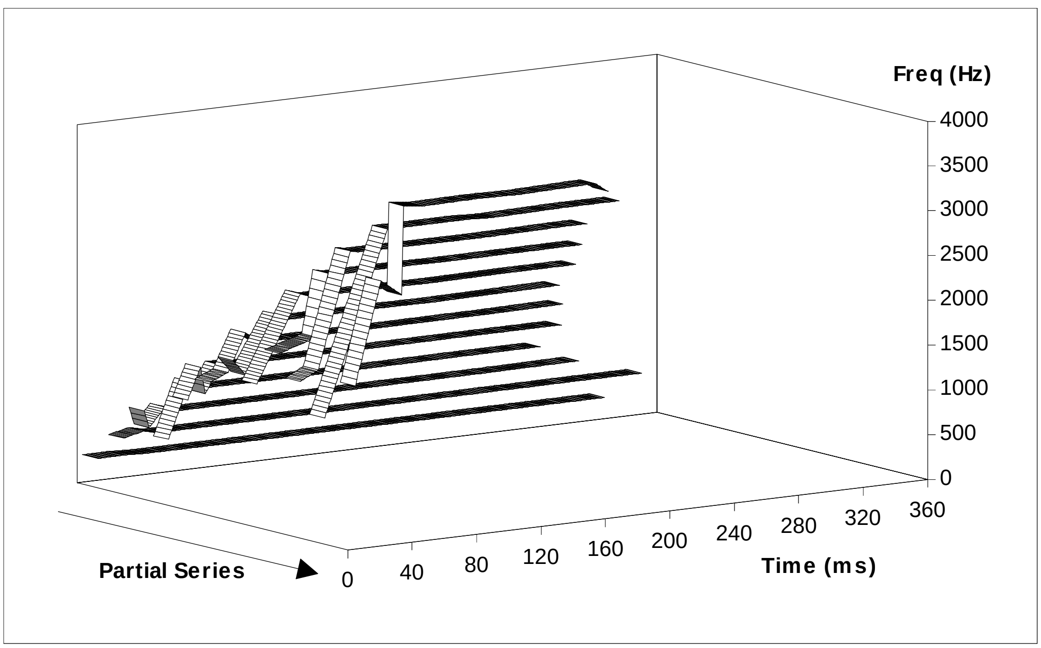

Figure 1.4 Frequency Envelopes for Grey’s Trumpet

Just as $F_x[n]$ of x has an upper bound $f_{max}(x)$, so it has a lower bound of $f_{min}(x)$=min($F_x[n]$). The interpretation of Nyquist’s theorem can be extended to give eqn. (1.4) if, conceptually, x is modulated up to $f_{min}(x)$ from a baseband spanning from DC to

$(f_{max}(x)$- $f_{min}(x))$. This results in a much lower $f_{opt}(x)$ because it is unusual for $F_x[n]$ to extend frequency modulation down to DC in music synthesis. Instead eqn. (1.4) optimises $f_{opt}(x)$ to the modulation range required by x irrespective of operating frequency resulting in large potential savings for high frequency partials with a narrow $F_x[n]$ modulation range. To generalise, for a note with α=*max pitch* / *min pitch* and pitch-invariant partial frequency ratios, $f_{min}(x)= f_{max}(x)/\alpha$. For instruments with a low pitch modulation (e.g. a piano), $\alpha \cong 1$ implying that $f_{opt}(x)$ will be low for all x. However, fundamental physical properties of musical signals mitigate against such a direct optimisation.

$$f_{opt}(x) = 2(f_{max}(x) - f_{min}(x)) \tag{1.4}$$

The Heisenberg inequality, given in equation (1.5), states that the product of time (Δ_t) and frequency (Δ_f) resolution has an upper bound (Gabor, 1947), which can be interpreted in an AS context as the observation that envelope control rate has an inherent proportionality with oscillator operating frequency. Alternatively, the inequality becomes an approximate equality. High frequency partials, such as transients, possess features at a fine time resolution and require a rapid control rate. Conversely, low frequency partials vary relatively slowly with time and may exploit a lower time resolution. Such reasoning also forms the basis for the new discipline of 'wavelet' theory (Rioul and Vetterli, 1991). A frequency domain interpretation is that control rate is proportional to the width of modulation sidebands generated by $A_i[n]$ (less so by $F_i[n]$) giving a 'constant-Q' time-frequency paradigm (see section 3.1.4). For AS, the critical sampling of eqn.(1.3) encapsulates these principles whereas eqn. (1.4) does not.

$$\Delta_t \Delta_f \geq \frac{1}{4\pi} \tag{1.5}$$

However, eqn. (1.4) is valid for the exception to the general rule when the left-hand side of eqn. (1.5) is much greater than the right-hand side as is the case when high frequency partials have a slow control rate and little transient character. For these reasons, it can be understood that eqn. (1.4) can be used to estimate $f_{opt}(x)$ if $\{f_{max}(x), f_{min}(x)\}$ is determined

- more generally - from an estimate of the bounds of the time-evolving spectral mainlobe of x under modulation of $A_i[n]$ and $F_i[n]$ such that aliasing of external components does not degrade perceived synthesis quality.

1.4.3 A Proposal for Multirate Additive Synthesis

Returning to the realities of VLSI development, as other architectures have fallen in and out of favour, the TOB form has remained immune - due to the *scalability* of AS and the fact that increasing clock rates merely increases S. It is a proven starting point for the work presented in this thesis. If each sinusoid x in an oscillator bank has an independent, optimal sample rate $f_{opt}(x)$, then many different sample rates must be supported simultaneously. Such a digital signal processing system is classified as "multirate" (Crochiere and Rabiner, 1983) and hence Multirate Additive Synthesis (MAS) is used as a label in later discussions. It is the purpose of this thesis to establish (i) a theoretical MAS paradigm, (ii) how it is subject to practical constraints, (iii) to derive some preliminary estimates about performance improvement as compared to classical AS and, finally, (iv) how MAS may be mapped into an efficient application-specific VLSI architecture.

The factors involved with determining $f_{opt}(x)$ in the context of note-based AS have been identified, but two further issues are raised which are pertinent to mention here. The first is that synthesis of x at $f_{opt}(x)$ creates sidebands around harmonics of $f_{opt}(x)$ which lie within the audio spectrum because $f_{opt}(x)<f_s$, representing audible time-quantisation noise and lending an unpleasant 'digital' quality to synthetic sound. Signal interpolation is required to raise the sample rate from $f_{opt}(x)$ to f_s to suppress this noise. Though an expensive process, interpolation can be applied to a sum of oscillators such that net savings in exploiting $f_{opt}(x)$ exceed the overhead. The second issue is one of multirate oscillator scheduling. The TOB executing at a single rate of f_s requires only a simple 'round-robin' schedule. Supporting S unique incommensurate $f_{opt}(x)$'s in real-time poses a considerable challenge and quantisation to a finite set of schedulable $f_{opt}(x)$'s is necessary.

1.4.4 Thesis Structure

In Chapter 2, the context and development of AS are reviewed. Firstly, there is a brief summary of alternative synthesis techniques which highlights the excellence of AS as outlined in section 1.1.1. Secondly, the development of digital AS and spectral modelling is traced building upon section 1.3.1. A literature survey of proposed AS optimisation methods is presented which stresses the importance of minimising computation without reducing AS generality. Thirdly, the use of the IFFT in emulating a TOB is documented in detail because it is emerging as one of the most popular and economic methods for implementing AS to which MAS offers an alternative solution.

A paradigm for MAS is developed in Chapter 3 from first principles. After a brief review of the basic operations of multirate DSP in the context of the sinusoidal synthesis, it is demonstrated how AS is transformed into MAS by the introduction of multirate filterbanks in time-domain that relate to specific subband decompositions in frequency domain. The application of various decompositions to expected frequency distributions of partials in note-based music is discussed and concludes with a cost / benefit analysis of MAS. Arising from this, a paradigm based on QMF filterbanks is proposed which uses a hierarchical subband decomposition.

Practical details of QMF filterbank implementation are discussed in Chapter 4. Both FIR and IIR designs are possible and each has attendant advantages and disadvantages. For functional transparency in MAS, a number normalising modifications to the QMF structure are specified. In particular, the phase-distortion inherent in QMF filtration is of concern. Chapter 5 identifies solutions to the problem of non-infinitesimal transition widths in QMF filterbanks. A method for their logical exclusion is presented and discussed. In the light of the shortcomings of QMF filterbanks, a MAS-specific design is presented - the Physical Exclusion Filterbank (PEF) - and supported by simulation results. Finally, the range of filterbank options, and their application to MAS, are summarised.

To reach some speculative conclusions about the performance improvement afforded by MAS, Chapter 6 profiles an experimental simulation of the resource allocation of a hypothetical, yet feasible large-scale practical MAS application: the synthesis of a symphony orchestra with one AS voice per instrument. The experiment involves real musical scores and instrument timbres. The resulting allocation histograms are applied to models that provide benchmarks of the speedup and cost of MAS for a given filterbank configuration. Another important aspect of the chapter is that the experimental scenario provides a context for discussing many of the higher-level, application-oriented issues of MAS in contrast to its low-level computation.

A key factor in the design of a successful MAS implementation is an efficient multirate sinusoidal oscillator bank. Digital sinusoid generation is a relatively mature subject and thus a significant proportion of Chapter 7 is devoted to a literature review of current techniques, with a bias on their efficacy for AS / MAS. Oscillator designs are classified under two headings as; (i) recursive and (ii) phase-accumulator based. The use of the CORDIC algorithm for complex sinusoid generation is investigated and results are presented. However, it is evidenced that a modification to the traditional look-up table (LUT) approach has best performance in terms of throughput / silicon area.

In Chapter 8, a systems analysis of a hypothetical MAS synthesiser indicates the utility of mapping MAS into an ASIC-based coprocessor (MASC) which forms part of a multiprocessor with general purpose CPUs executing software processes. Standard computer engineering concepts are applied in a novel context. The installation of a MASC within a multiprocessor topology is discussed and leads to the proposal for inter-processor communication via shared memory, which isolates high control bandwidths from the software processes via a MAS-specific memory hierarchy. When combined with an object-oriented data structure design, the fundamental basis of MASC thoughput optimisation by interleaved burst processing is acheived.

Chapter 9 completes the MASC functional specification by specifying all of the remaining higher-level scheduling and synchronisation algorithms and data structures that are necessary for MASC operation in the context of a multiprocessor MAS synthesiser. A quantitative review of expected MASC performance as a function of its design parameters (and in the light of Chapter 6) shows that MAS optimisation is manifest in the MASC by a reduction in the required internal clock rate. Finally, a data-flow model of the MASC and its major functional units is provided, leaving the way open for subsequent behavioural simulation and prototype implementation.

1.4.5 Publications Arising from Thesis Research

- Conference Publications:- (Phillips et al, 1994), (Phillips et al, 1996a)
- Journal Publications:- (Phillips et al, 1996b)