A Multirate Optimisation for Real-Time Additive Synthesis

Desmond PHILLIPS
Desmond.Phillips@dur.ac.uk
Durham Music Technology*
School of Engineering, University of Durham, Science Laboratories,
South Road, Durham City, County Durham, DH1 3LE, UK.

Alan PURVIS
Alan.Purvis@dur.ac.uk

Simon JOHNSON
Simon.Johnson@dur.ac.uk

Abstract

The optimisation of a large set of digital sine oscillators is considered. The proposed solution is highly transparent and works by exploiting redundancies inherent in real-time additive synthesis. A scheme of multirate interpolation "harnesses" and oscillators using complex representation is introduced. It is concluded that a potential exists to synthesise typical spectra with a "speedup" factor approaching one order of magnitude when compared with an unoptimised implementation.

1. Introduction

The theory of additive synthesis using a large set of continuously and arbitrarily controllable sinusoids has an ancestry as old as Computer Music. The direct mapping from frequency to time domain permits an absolute level of control over timbre that is unequalled in competing synthesis techniques. However, its theoretical attraction is often disregarded when the high cost of practical implementation is considered. In real-time, thousands of oscillators must operate concurrently; each also requires two high-resolution streams of data for dynamic control of frequency and amplitude, which themselves must be synthesised at oscillator sample rate.

High performance synthesizers constructed with this "brute force" approach make useful research platforms [Jansen, 1991]. They provide the synthesis resources to enable the development of real-time software environments using spectral modelling. By this means, the power of additive synthesis is becoming increasingly accessible to musicians and composers. Cost is a secondary consideration.

Commercial products using additive synthesis confine themselves to limited applications where convenient economies can be made. An example is pipe organ synthesis [Comerford, 1993]. Coincident harmonics are overlaid in single oscillators, and steady-state conditions use wavetable synthesis. Large savings are made in the number of oscillators required. However, generality is necessarily sacrificed for cost.

Recent research has focused on exploiting the efficiency and generality of the Inverse Fast Fourier Transform (IFFT); a direct transformation from frequency to time domain. A major disadvantage is that it is discontinuous. Indirect use of the IFFT by Short-Term Fourier techniques goes some way to overcoming this obstacle [Rodet and Depalle 1992], but not without side-effects of fitting the IFFT to a generalised additive synthesis model.

A fourth strategy for tackling the problem of optimising additive synthesis, is to start with an analysis of the way it is computed. Is it possible to refine the "brute force" approaches to achieve the efficiencies offered by IFFT techniques?

2. Redundancies

The key to increasing efficiency is to eliminate redundant computation. We identify four specific areas of interest:

1. Optimal Oscillator Sample Rate. A sine oscillator producing a tone of 1kHz requires a sample rate of at least 2kHz: the Nyquist rate. However, it would be customary to update it at e.g. 44.1kHz: a potential performance gain of x22. A major drawback is that the optimal sample rate of a sine oscillator is a function of its output frequency, arbitrary and time-varying. The scheduling and interpolation overheads for thousands of incommensurate sample rates is of absurd complexity (akin to converting square waves to sinusoids). However, a near-optimal solution is possible with realistic costs.

2. Real Vr Complex Representation. Sampled real signals exist as symmetrical sidebands about harmonics of the sample rate. This in fact
3. Frequency Range. To assume that an oscillator always requires a sample rate of \( \geq 40kHz \) is to assume that it needs to operate over the full 20Hz-20kHz range during its allocated lifetime. For additive synthesis in music, an oscillator naturally restricts itself to a limited range about a central frequency. We can use this information to assign a fixed sample rate \( < 40kHz \), as long as we provide sufficient bandwidth for vibrato effects.

4. Latency. Real-time systems are often classified as 'hard' or 'soft'. It is a rare property of any real-time synthesis algorithm that it be perceptually 'hard': a condition satisfied if the response to an event is a fixed constant not more than for example 10ms. This time interval is a qualitative redundancy that may also be exploited.

3. The Harness Concept

A systematic exploitation of these redundancies can be made within the process of integrating a large set of sine oscillators into a single data stream. To do this, we introduce the idea of a multirate interpolation 'harness' which formalizes the natural grouping of oscillators that occurs in the additive synthesis of music.

A harness is a set of fixed, near-optimal sample rates to any oscillator and interpolates its output up to the Nyquist rate for digital audio of \( \geq 40kHz \). For a single oscillator, this would be highly inefficient, but a harness is a linear system that will also interpolate sumped oscillator sets. The more oscillators applied to a harness, the greater the efficiency because harness computation is a fixed overhead. In practice, it is envisaged that the number of harnesses required is considerably less than the number of oscillators (e.g. \( 10 \) to \( 10,000 \)).

Dataflow in classical additive synthesis displays a process of refinement from a large oscillator set down to a single data stream. In the middle layer of this hierarchy, the data bandwidth is greatly reduced with a small number of logically separate sample streams. For example, string, brass and woodwind sections may have different post-processing requirements (reverb, pan etc.) before final mixing to 16-bit stereo at 44.1kHz. Applying a tanh to each stream fits in with this paradigm without imposing crippling restrictions.


Within a harness, the audio spectrum is decomposed into a multirate progression of \( n_{oct} \) octaves [Bell-Hajek, 1989]. Sample data cascades from the lowest octave, \( \#1 \), to the highest, \( n_{oct} \). Sample rate doubles from octave to octave with a half-band filter providing interpolation up to the final Nyquist sample rate. By this means geometric progression of allocatable oscillator sample rates is created. Using a priori knowledge of an oscillator's output frequency range, a run-time allocation algorithm assigns it to the lowest possible octave so that its output cannot alias. Hence it is possible to provide a sample rate close to the 'ideal' for an oscillator.

Figure 1 illustrates the dataflow of a prototype harness. All signals are complex rather than real to exploit a half-band oscillator update rate: \( X(1, \#1) \) denotes summed oscillator sets allocated to octaves. The cascade starts with a special stage for sets \( X(1) \) & \( X(2) \), then a generic intermediate stage for \( X(1) \times 2 \times \#n_{oct} \) and then a final stage. Interpolation of complex signals uses a pair of identical half-band filters operating in parallel on real and imaginary components.

![Figure 1: Prototype Harness Dataflow](image)

The cascade form of harness is attractive for at least two reasons. First, the partitioning of the audio spectrum into octaves is a natural paradigm for musical applications. Second, it is regular, efficient to compute and has a simple dataflow topology. As a multirate system, throughput is dominated by the highest sample rate processes: converging to a limit with increasing oct.

High order half-band filters are required for a narrow transition width and good stop-band attenuation. High-order filters incur a significant latency. Maximum harness latency increases exponentially with \( n_{oct} \), due to the low sample rates of the lowest octaves: it must not exceed 10ms. However, harness optimality converges to a limit with a low value of \( n_{oct} \). Hence
we choose a de facto value of $n_{oct}$=d for our present cascade harness designs.

Differential latencies between octaves are corrected by introducing delay lines to the inputs of higher octaves to prevent them propogating faster than lower octaves. Latency is equalised to the delay of the longest path from the lowest octave.

3.2 Frequency Domain Operation

A harness has $n_{oct}$ stages with outputs denoted $Y_{(j)}=j\cdot c_{j} \cdot c_{j+1}^{n_{oct}}$. Figure 2 shows the mixing of the two lowest octaves. $X(1)$ and $X(2)$ are each interpolated by a half-band filter $H_a$. $X(1)$ is down-shifted by $\Omega_1$ and $X(2)$ is up-shifted by $\Omega_2$. They are then summed to make $Y(1)$.

$$X(1) \rightarrow Y(1)$$
$$X(2) \rightarrow H_a$$
$$\Omega_1=-0.2\pi$$
$$\Omega_2=+0.2\pi$$

Figure 2: First Harness Stage

Intermediate stages ($1<n_{oct}$) are added using instantiations of the generic stage as in Figure 3. We take advantage of the lower relative bandwidth occupied by previous octaves by interpolating $Y_{(j-1)}$ with $H_b$, a lower order filter. $Y_{(j-1)}$ is shifted to the left of $DC$ by a down-shift of $\Omega_1$, but $X(1)$ is up shifted by $\Omega_2$. As a result, their sum $Y_{(j)}$ is also symmetric about $DC$.

$$Y(1) \rightarrow H_b$$
$$X(1) \rightarrow Y(1)$$
$$H_a$$
$$\Omega_1=-0.3\pi$$
$$\Omega_2=0.3\pi$$

Figure 3: Intermediate Harness Stage

Figure 4 illustrates the final stage of the harness. $Y_{(n_{oct})}$ is interpolated by $H_b$ and up-shifted by $\Omega=3\pi$. A component of $Y_{(n_{oct})}$ can be dissipated to generate a single signal as there is no longer any negative frequency component.

An important factor in harness design is the ratio of $H_a$ stopband width to its cutoff frequency. A figure of 3:8 emerges as a good compromise between filter order, latency and usable bandwidth. FRs emerge as the best filter option because of their fixed latency and fixed-point arithmetic. The single advantage of high-order IIRs, relative to FRs, is that they exhibit lower latency.

The prototype harness specification is thus: $N=67$ for $H_a$ and $N=35$ for $H_b$, an output sample rate of 66.67kHz at 100dB S/N ratio, a multiplier bandwidth requirement of 5.33MHz, and a constant latency of 5.75ms. It is presently simulated in C on a SUN SPARC workstation.

3.3 Spectral Partitioning

We can take advantage of frequency shifts within the harness to develop a flexible strategy for mapping octaves into the audio spectrum. Figure 5 shows the spectral partitioning of the prototype harness. Octaves are overlapped in order to permit a working figure of at least 20% change in an oscillator's central frequency. The bandwidth of each octave is a fraction of its sample rate (0.58) and hence there exists a motive for minimizing it. Another reason for overlapping is to accommodate bands of noise in addition to sinusoidal partials.

It proves to be a good compromise between a 'collapsed' schema (where all octaves include DC) with higher sample rates, and an 'extended' schema with lower sample rates but no vibrate capability. As octaves are 'extended', positive side-effects occur; (1) the order of $H_b$ drops and compensates for increasing peak harness sample rate, (2) latency drops dramatically due to the interaction of these two factors.

Sound synthesis techniques
3.4. Analysing Optimality

Harnessing is a statistical optimisation dependent on the time-evolving spectral distribution of oscillators. Research is in progress to obtain authoritative figures of its optimality. However, we can observe some trends as illustrated in Figure 6. It shows the ratio of harnessed oscillator updates required for the ascending C's of a piano (with a variable length series of partials) relative to unharvested oscillators updating at 44.1kHz.

Figure 6: The Relative Cost of Harvested to Unharvested Additive Synthesis

The cost is bounded by the values -0.09 and 0.26. Often, 20 to 30 harmonics suffice in additive synthesis (Moore, 1977). C3 (middle C) has a cost ratio of -0.11 with harvesting of 22 harmonics. Using these observations, an aggregate cost of -0.15 is suggested indicating an overall performance gain for harvesting of approximately x7.

4. Oscillator Design

The implementation of a large oscillator set is a side-issue of fine-grain data parallelism. However, harnessing presupposes multiple complex oscillators. It introduces two complexities, which have simple solutions.

4.1. Complex CORDIC Oscillators

Complex oscillators can be implemented efficiently using the CORDIC vector rotation algorithm (Hu, 1992). It can be visualised as a nest of polygons with sides converging towards a circle. Successive approximations of θ leads to a path through the polygons converging on a point (x,y) on the circle from a starting vector of α. As k=1.647, and φ/α are orthogonal, we have independent control of oscillator phase, frequency and magnitude. It is economic in VLSI (in additions are required) and can easily be pipelined. Compare with the alternative of two sine look-up table accesses and two multipliers.

If it is a signal then we can use CORDIC to shift its spectrum (central about DC) up to a desired central frequency. By this means, it is possible to introduce noise components into the harness, an important adjunct to the synthesis of stochastic partials. We will take advantage of near-optimal sample rate but the shifted spectrums must remain confined to the bands set out by spectral partitioning.

4.2. Oscillator Scheduling

An allocation model for scheduling multirate oscillators involves the use of λ frames repeating at the octave #1 rate with 2π/λc^2 update slots. λ. 32 points are needed for λ=2. A frame can support any combination of oscillators such that the combined sum of updates does not exceed 2π/λc^2 e.g. 4 x octaves λ or 2 x 2s, 2 x octave #3 oscs or 1 octave #4 osc. Given an oscillator's octave, a simple sequential first-fit allocation algorithm through λ frames guarantees to make optimal use of resources.

5. Summary

Harnessing offers a performance increase of x7 by exploiting (1) near-optimal sample rate, (2) complex representation and (3) spectral partitioning. It fits transparently into an oscillator-set model of additive synthesis and its associated algorithms and architectures. Simple, deterministic DSP techniques are used. Cost-effective VLSI implementation appears highly feasible.

References


The authors would like to acknowledge the financial support provided for this paper by EPRSG and the School of Engineering, University of Durham.