Fast Additive Sound Synthesis Using Polynomials

Matthias Robine, Robert Strandh and Sylvain Marchand

firstname.name@labri.fr

Université Bordeaux 1, LaBRI, SCRIME

Plan

Plan

Sinusoidal Modeling, Additive Synthesis

Fourier's theorem: "*any periodic function can be modeled as a sum of sinusoids at various amplitudes and harmonic frequencies.*"

McAulay-Quatieri representation:

$$a(t) = \sum_{i=1}^{N} a_i(t) \sin(\phi_i(t))$$

with:

$$\begin{cases} \frac{d\phi_i}{dt}(t) & = & 2\pi f_i(t) \\ \phi_i(t) & = & \phi_i(0) + 2\pi \int_0^t f_i(u)du \end{cases}$$

Sound Parameters

- Parameters of partials:
  amplitude, frequency, initial phase
- Slow evolution of the parameters
- We consider parameters constant for a short length:

$$a(t) = \sum_{i=1}^{N} a_i \sin(2\pi f_i t + \phi_i)$$

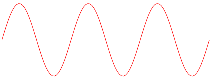
Plan

Modeling Parameters

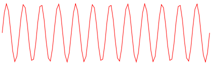
Polynomials to model the parameters of the sinusoidal model:

- McAulay & Quatieri 1986: 3^{rd}-degree polynomials to model the phase parameter,
- Ding & Qiang 1997: 4^{th}-degree polynomials to model the phase parameter,
- Raspaud & al 2005: *Poly-Sin* model, polynomials to model the slow variations of the sound parameters.

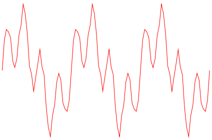
Sum of Sinus Functions



$$a_1(t) = a_1 \sin(2\pi f_1 t + \phi_1)$$

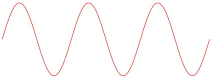


$$a_2(t) = a_2 \sin(2\pi f_2 t + \phi_2)$$

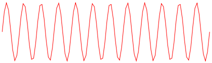


$$a(t) = \sum_{i=1}^{N} a_i \sin(2\pi f_i t + \phi_i)$$

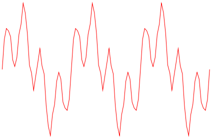
Sum of Polynomials



$$P_1(t) = \sum_{i=0}^{d} \alpha_i t^i$$



$$P_2(t) = \sum_{i=0}^{d} \beta_i t^i$$



$$P(t) = \sum_{i=0}^{d} (\alpha_i + \beta_i + \ldots) t^i$$

Plan

1. Additive Synthesis
2. Using Polynomials
   - Modeling Parameters
   - Fast Sound Synthesis
3. PASS (Polynomial Additive Sound Synthesis)
   - Approximation of a Partial
   - Time Events
   - Change of Parameters
   - Complexity and Results

Polynomial Approximation of the Signal of a Partial

Unit signal $u(t)$ with $a = 1$, $f = 1$, $\phi = 0$: $u(t) = \sin(2\pi t)$
approximated by unit polynomial $U(t)$.

We have to choose:

- the approximated part p of the period of the signal,
- the polynomial degree.

Finding the Best Coefficients

Quality of the approximation evaluated with the SNR:

$$\mathrm{SNR} = 10 \cdot \log_{10} \left(\frac{\int_0^p u^2(t)dt}{\int_0^p (u(t) - U(t))^2 dt} \right)$$

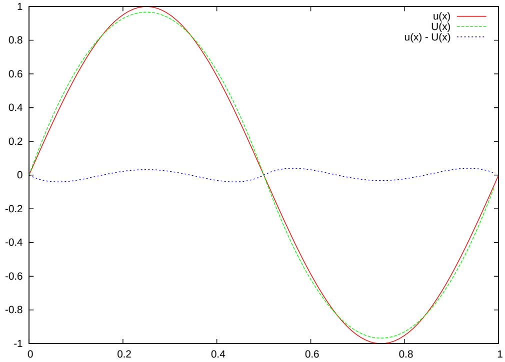
To find the best polynomial coefficients, for p and d given, we have to:

- maximize the SNR (minimize the denominator),
- maintain a piecewise continuity.

Noise

p	d	C^0 SNR (dB)	C^1 SNR (dB)
1/4	2	36	28
1/4	3	57	28
1/4	4	79	59
1/4	5	102	59
1/2	2	28	28
1/2	3	28	28
1/2	4	59	59
1/2	5	59	59
1	4	17	17
1	5	42	42

Error of Approximation



The noise adds some harmonics to the sinusoid.

Polynomial Generator

The polynomial P approximating the signal of a partial is given from the unit polynomial U:

$$P(t) = aU\left(ft + \frac{\phi}{2\pi}\right)$$

Polynomial Generator:

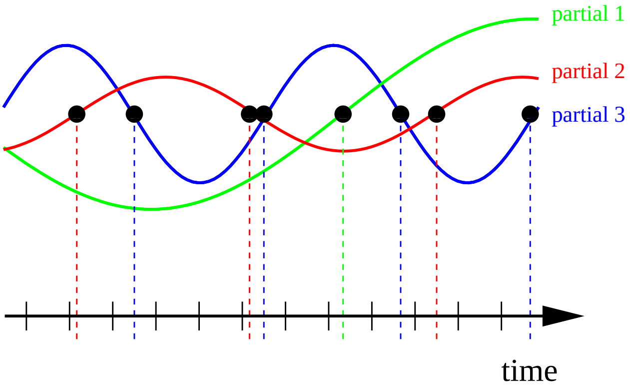
- sum of the polynomial coefficients from partials,
- values are samples of the sound.

Time Events

- Polynomial approximation of a partial is only valid on a length dependent on the frequency of the partial.
- Regularly, values of the generator must be updated, when an approximation is over.
- Regularly, the generator must be evaluated to produce a sound sample

We use a priority queue to manage the time events in PASS.

Time Events



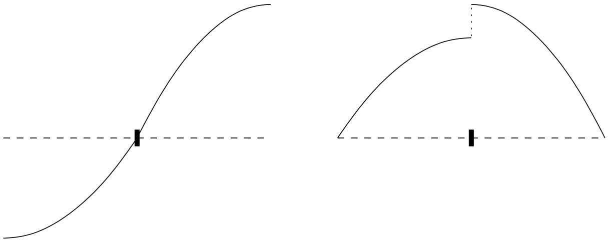
Update and evaluation events for the polynomial generator.

Change of Parameters

The sound parameters are considered constant on a short length, then they must be updated.

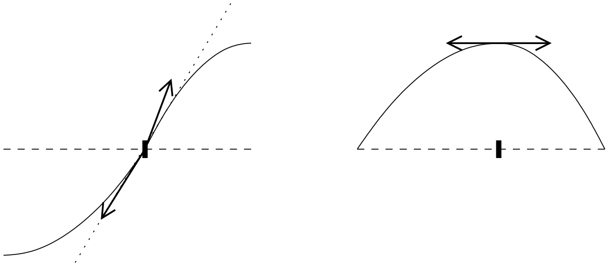
- Choice of the good moment to avoid discontinuities (clicks).
- New polynomial approximation.
- The generator must be updated.

Change of Amplitude



Best moment at the minimum of the signal to avoid discontinuity.

Change of Frequency



Best moment at the maximum of the signal to maintain C^1-continuity.

Complexity of PASS

Complexity C_1 for the managing of the time events, where $(P = 1/p)$ is the number of polynomials for the approximation of a partial:

$$C_1 = O\left(\left(\sum_{i=0}^{N} f_i\right) log(N)Pt\right) = O(N \log(N)\bar{f}Pt)$$

Complexity C_2, to produce a sound sample:

$$C_2 = O(dF_s t)$$

Then, with α an architecture-dependent constant:

$$C_{\text{PASS}} = O(\alpha C_1 + C_2)$$

Digital Resonator (DR)

Incremental algorithm:

$$\begin{cases} \Delta_\phi & = & \frac{2\pi f}{F_s} \\ s[0] & = & a\sin(\phi_0) \\ s[1] & = & a\sin(\phi_0 + \Delta_\phi) \\ C & = & 2\cos(\Delta_\phi) \\ \\ s[n+1] & = & C \cdot s[n] - s[n-1] \end{cases} \quad (1)$$

Complexity $C_{\mathrm{DR}} = O(NF_s t)$, with N partials in the sound.

Fast synthesis with fine control of the sound parameters.

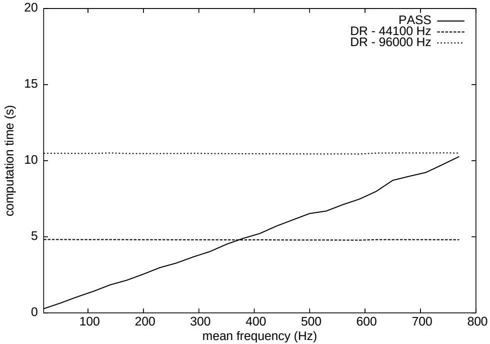
Comparison DR vs PASS

N	$\bar{f}$	F_s (Hz)	DR	PASS
4000	300	22050	**3.2** s	6.6 s
4000	300	44100	**6.3** s	6.6 s
4000	300	96000	13.7 s	**6.6** s

5 seconds of sound synthesis,
with an Intel Pentium 4 1.8-GHz.

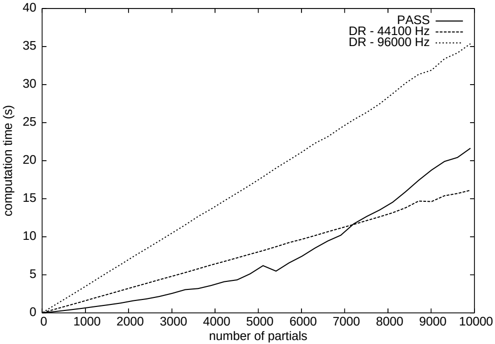
PASS method is not really affect by the sample rate.

Results: Function of the mean frequency



5 seconds of synthesis
on a PowerPC G4 1.25-GHz processor,
$N = 3000$.

Results: Function of the number of partials



5 seconds of synthesis
on a PowerPC G4 1.25-GHz processor,
$\bar{f} = 200$ Hz

Conclusion and Future Work

- Fast method of additive synthesis
- Fine control of the sound parameters
- Synthesis not really dependent on the sampling rate
- Efficient for low frequencies
- Could be mixed with DR with a frequency threshold