A GENERAL ANTIALIASING METHOD FOR SINE HARD SYNC

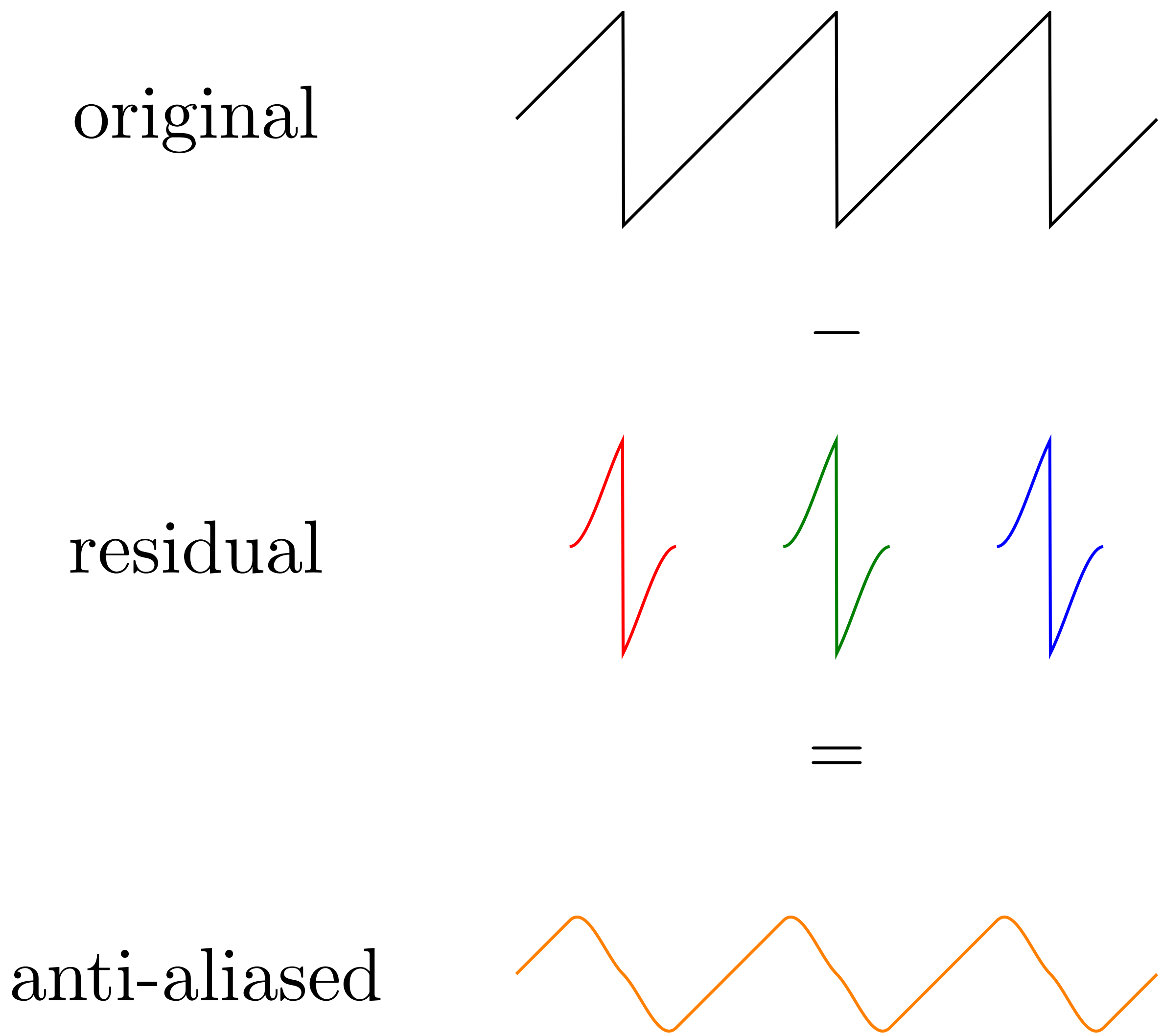
Pier Paolo La Pastina and Stefano D'Angelo

Orastron srl, Sessa Cilento, Italy

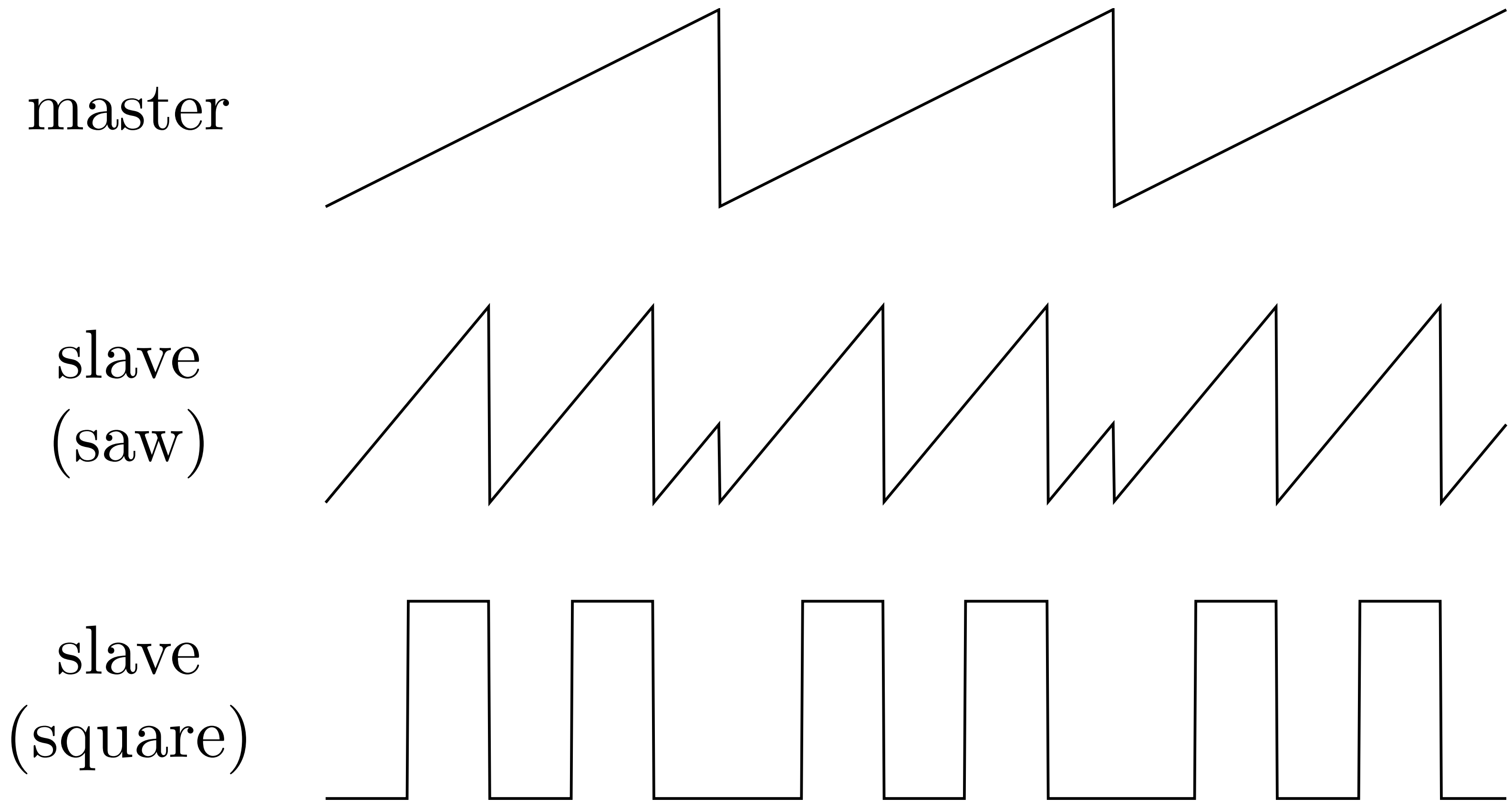
Vienna, Austria
Future Art Lab
7 September 2022

ORASTRON

BLEP

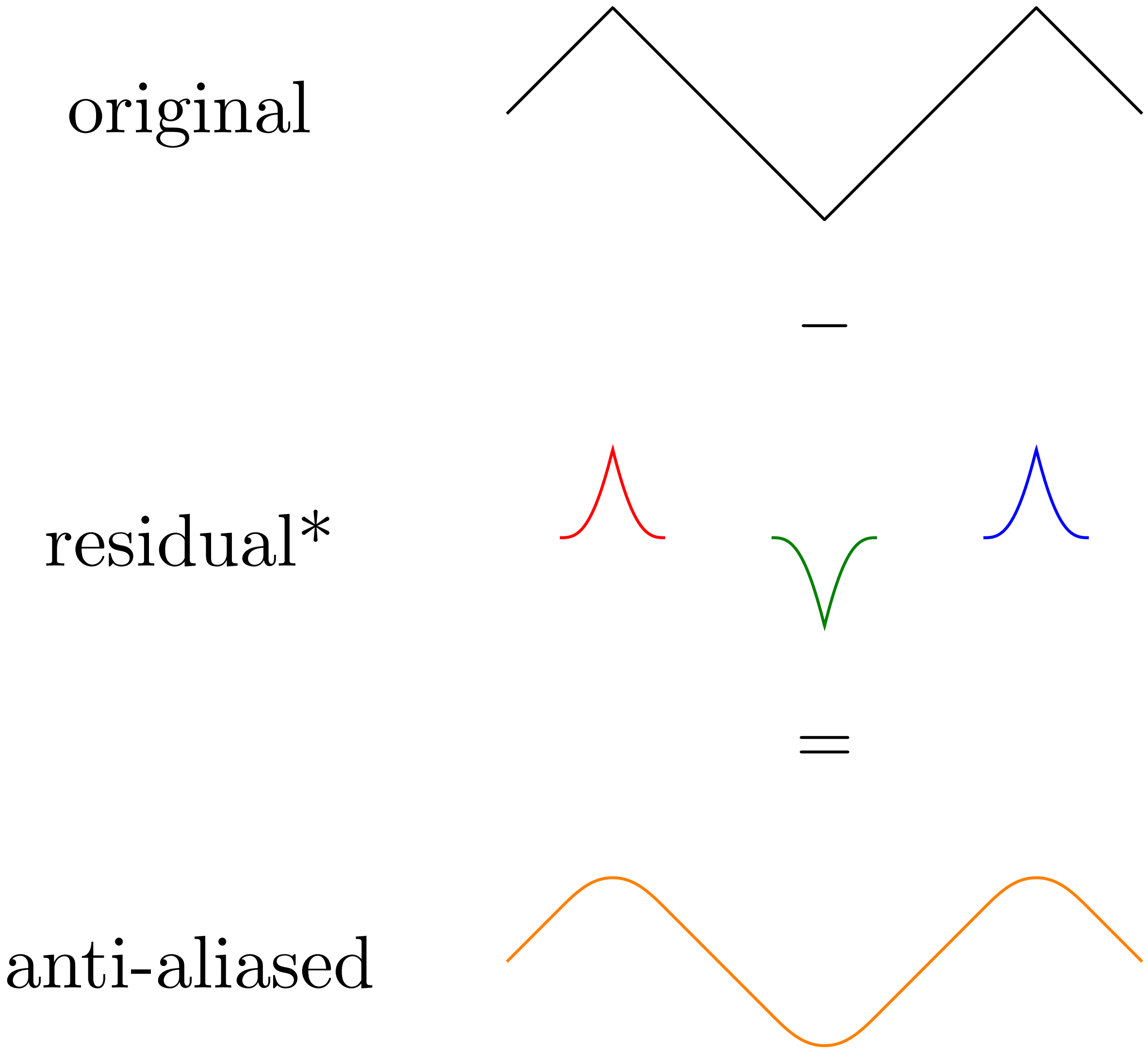


HARD SYNC

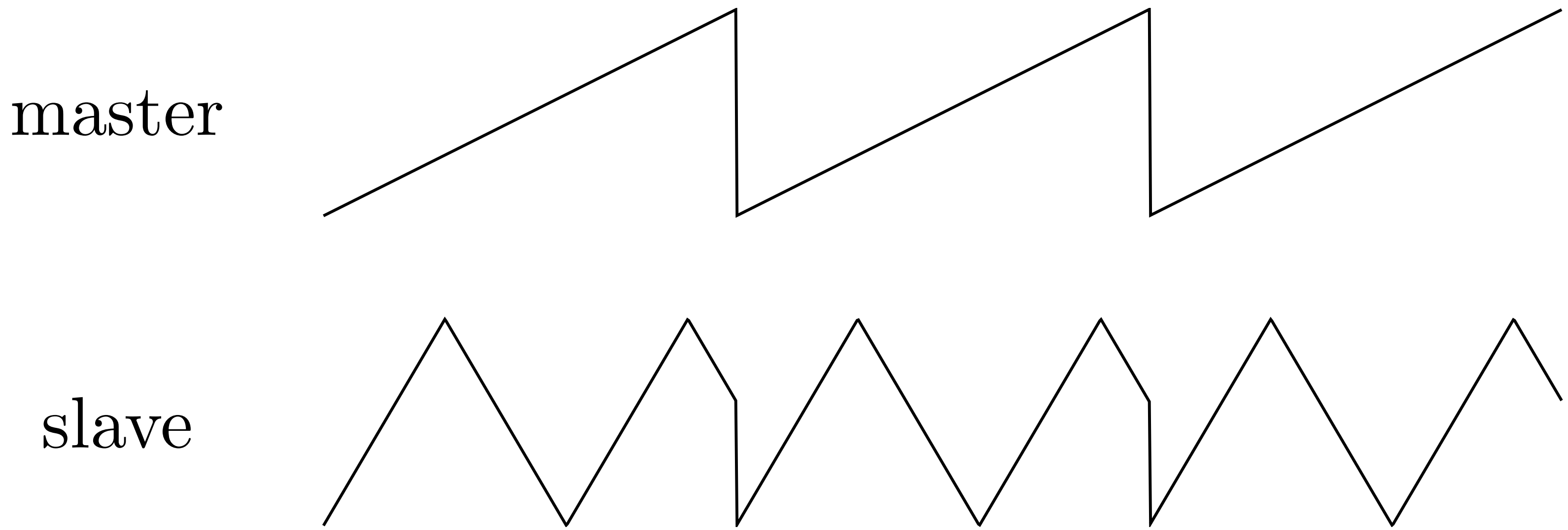


BLEP is enough

BLAMP

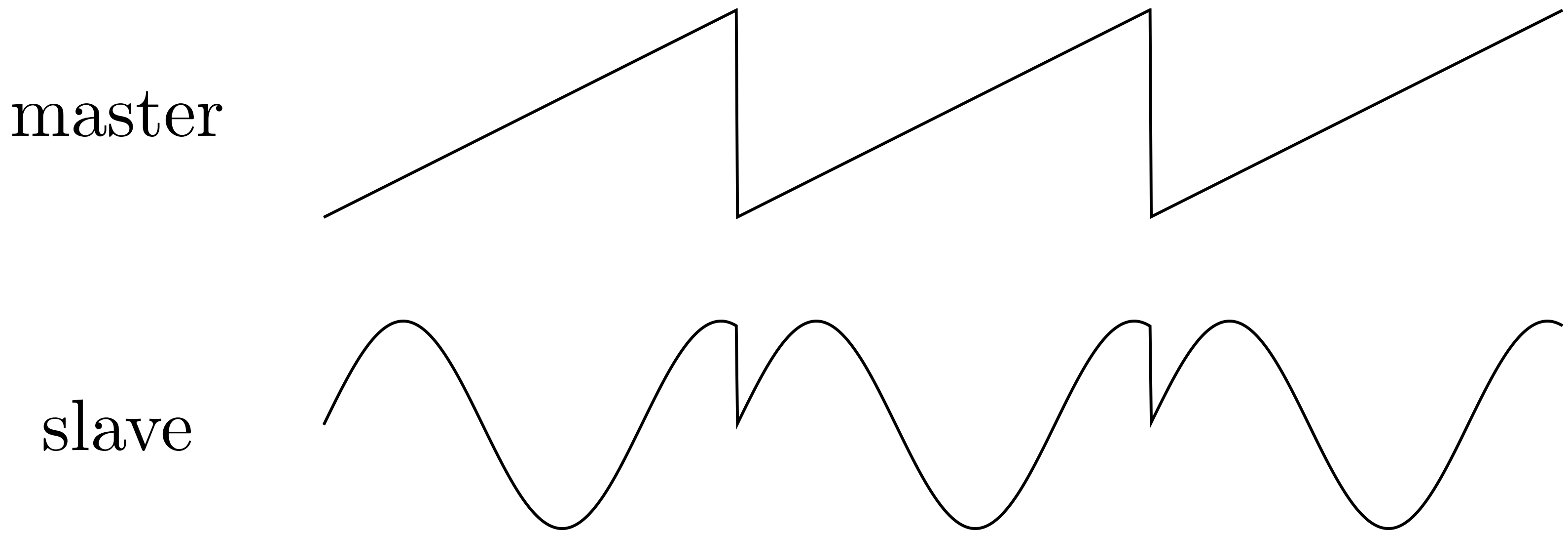


TRIANGLE HARD SYNC



BLEP + BLAMP

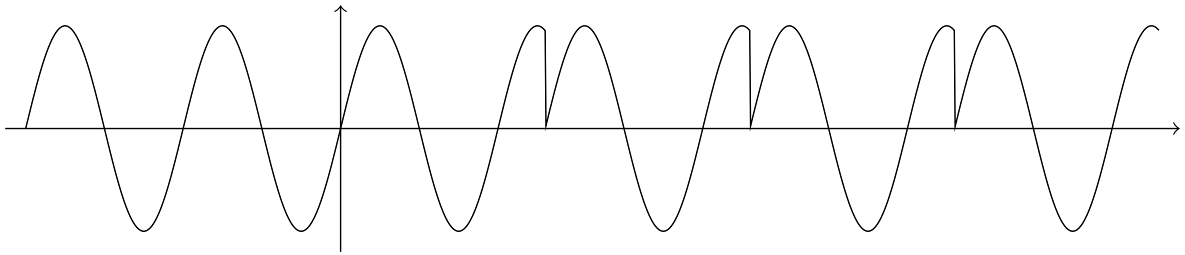
SINE HARD SYNC



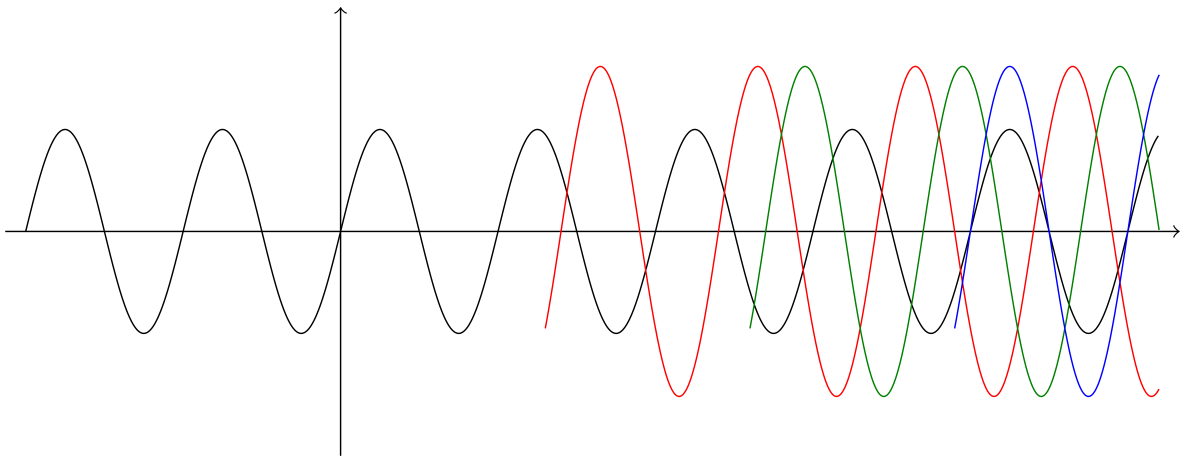
$$\frac{\mathrm{d}^n}{\mathrm{d}t^n}\sin\left(2\pi f t\right) = \left(2\pi f\right)^n \sin\left(2\pi f t + \frac{n\pi}{2}\right)$$

BLEP + BLAMP + infinite higher-order residuals

PROBLEM (RE)DEFINITION



$$x(t) = \begin{cases} \sin(\omega_0 t) & t \leq 0 \\ \sin(\omega_0 \bmod(t, T)) & t > 0 \end{cases}$$



$$x(t) = \sin(\omega_0 t) + \sum_{k=1}^{+\infty} f(t - kT)$$

$$f(t) = (\sin(\omega_0 t) - \sin(\omega_0 (t + T)))u(t)$$

$$= -2\sin\left(\omega_0 \frac{T}{2}\right) \cos\left(\omega_0 \left(t + \frac{T}{2}\right)\right) u(t)$$

$$y(t) = (h * x)(t) = \sin(\omega_0 t) + \sum_{k=1}^{+\infty} h(t) * f(t - kT)$$

assuming $h(t) * \sin(\omega_0 t) = \sin(\omega_0 t)$

ANTIALIASING FILTER

$$y(t) = (h * x)(t) = \sin(\omega_0 t) + \sum_{k=1}^{+\infty} h(t) * f(t - kT)$$

$$y(t) = x(t) + \sum_{k=1}^{+\infty} R(t - kT)$$

$$R(t) = g(t) - f(t)$$

$$g(t) = (h * f)(t)$$

1. if $h(t)$ is FIR, $R(t)$ has the same support of $h(t)$;

2. can $R(t)$ (or $g(t)$) be computed by a finite number of operations?

RESULTS

Polynomial and B-spline (piecewise):

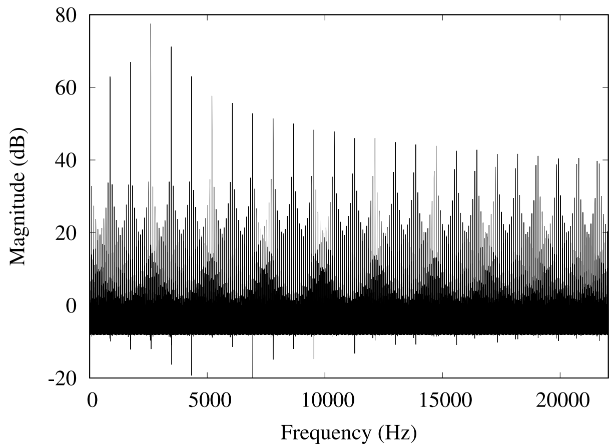
$$g(t) = p_0(\omega_0, T, t) + \sin(\omega_0 t) \sum_{i=1}^{P} p_{\mathrm{s},i}(\omega_0, T, t) + \cos(\omega_0 t) \sum_{i=1}^{P} p_{\mathrm{c},i}(\omega_0, T, t)$$

Trigonometric (sum of cosines):

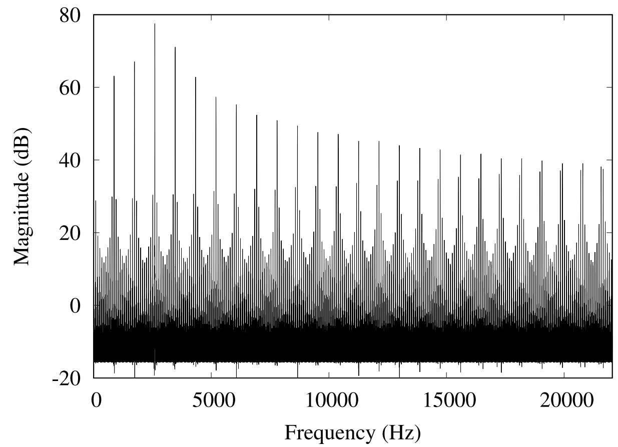
$$g(t) = k_0(\omega_0) \sin\left(\omega_0 \left(t + \epsilon + \frac{T}{2}\right)\right) + \sum_{i=1}^{N} k_{\mathrm{s},i}(\omega_0, T) \sin\left(\frac{i\pi}{\epsilon} t\right) + \sum_{i=1}^{N} k_{\mathrm{c},i}(\omega_0, T) \cos\left(\frac{i\pi}{\epsilon} t\right)$$

Potential optimization: Chebyshev polynomials

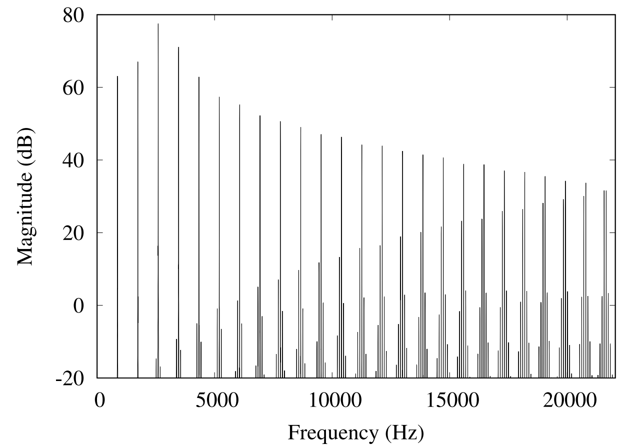
EXPERIMENT 1



No antialiasing



Frequency shifting method

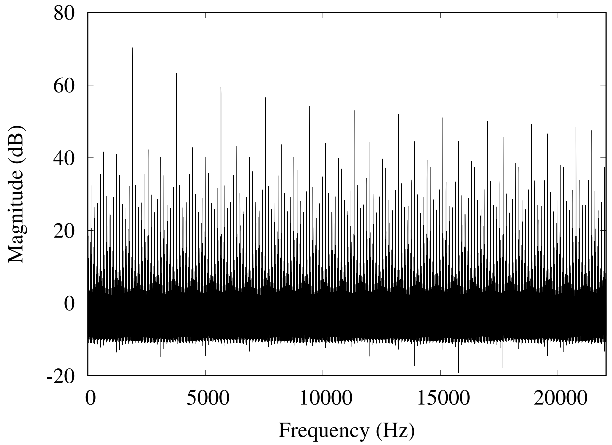


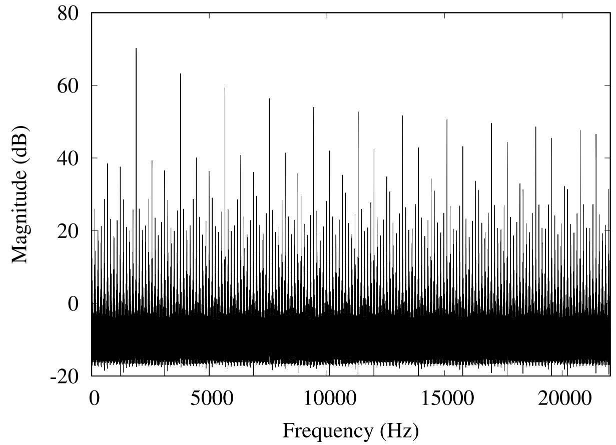
Proposed method

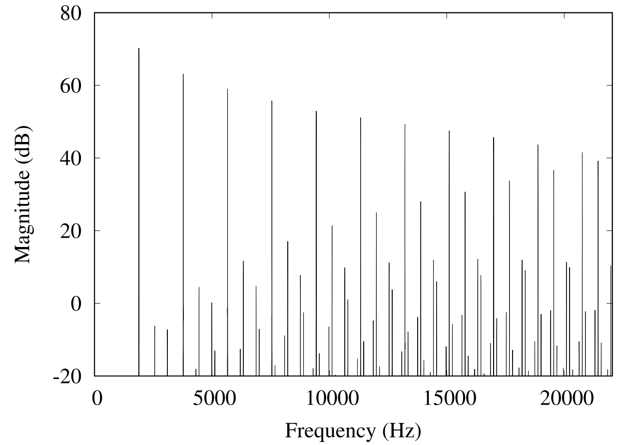
$f_\mathrm{s} = 44100$ Hz, master frequency 866.42 Hz, slave frequency 2900.33 Hz

FSM kernel: sinc w/ Kaiser window ($\alpha = 4$), 2 samples long
Proposed kernel: triangular, 2 samples long

EXPERIMENT 2







No antialiasing | Frequency shifting method | Proposed method

$f_\mathrm{s} = 44100$ Hz, master frequency 1888.10 Hz, slave frequency 517.88 Hz

FSM kernel: sinc w/ Kaiser window ($\alpha = 4$), 2 samples long
Proposed kernel: triangular, 2 samples long

CONCLUSIONS

- Better aliasing suppression than FSM when using same kernel length
- Lower computational cost than FSM when using same kernel length (polynomials + sine/cosine vs exponential integral evaluation)
- Explicit formulations given for polynomial, B-spline, and trigonometric kernels
- General methodology, any FIR kernel can be used (if you can cope with the math)
- The same approach could be applied to other waveforms, e.g., polygonal oscillators
- Grab the code at https://www.dangelo.audio/dafx20in22-sinesync.html