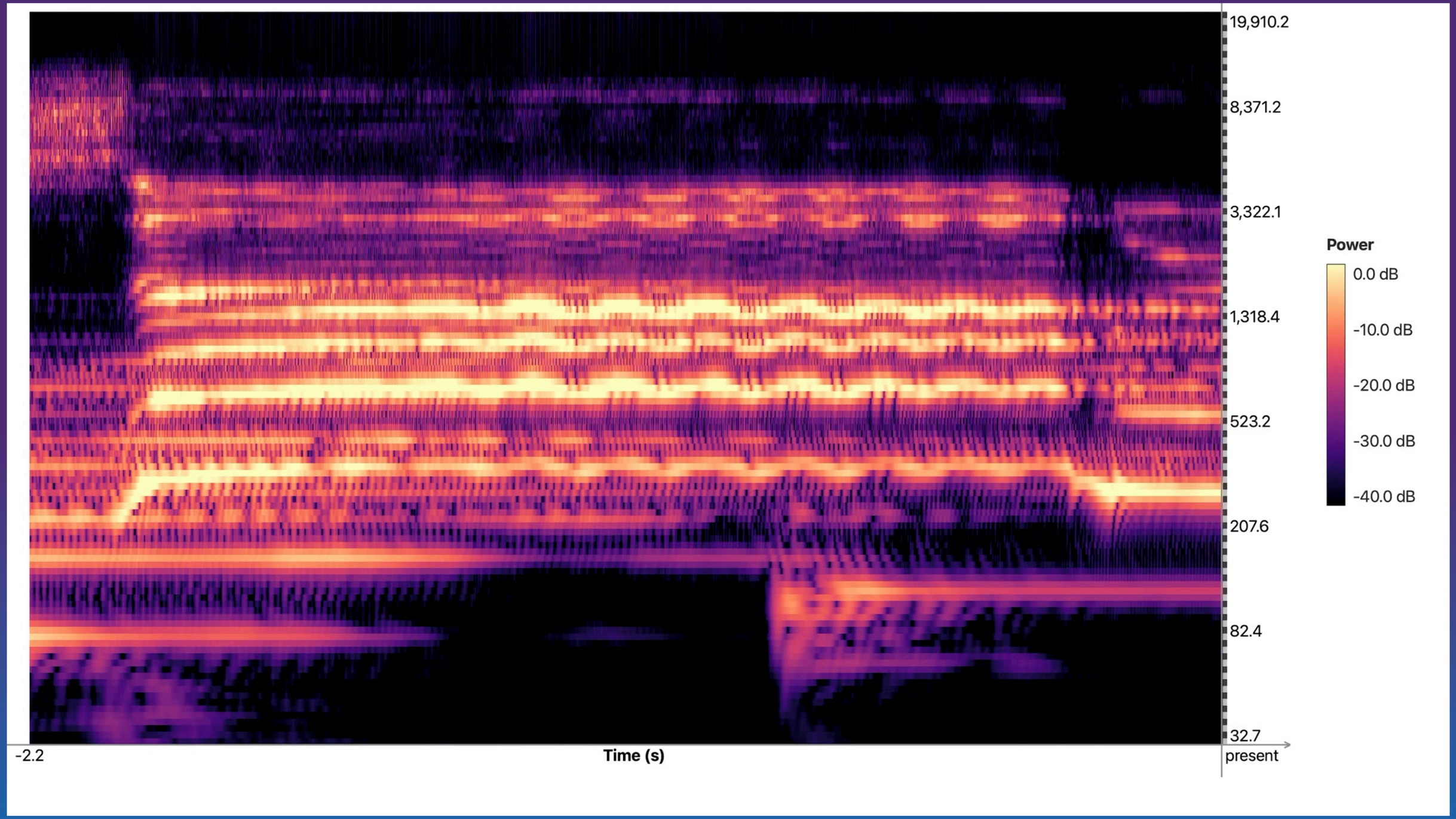
audio.dev
CONFERENCE
BRISTOL & ONLINE 2026

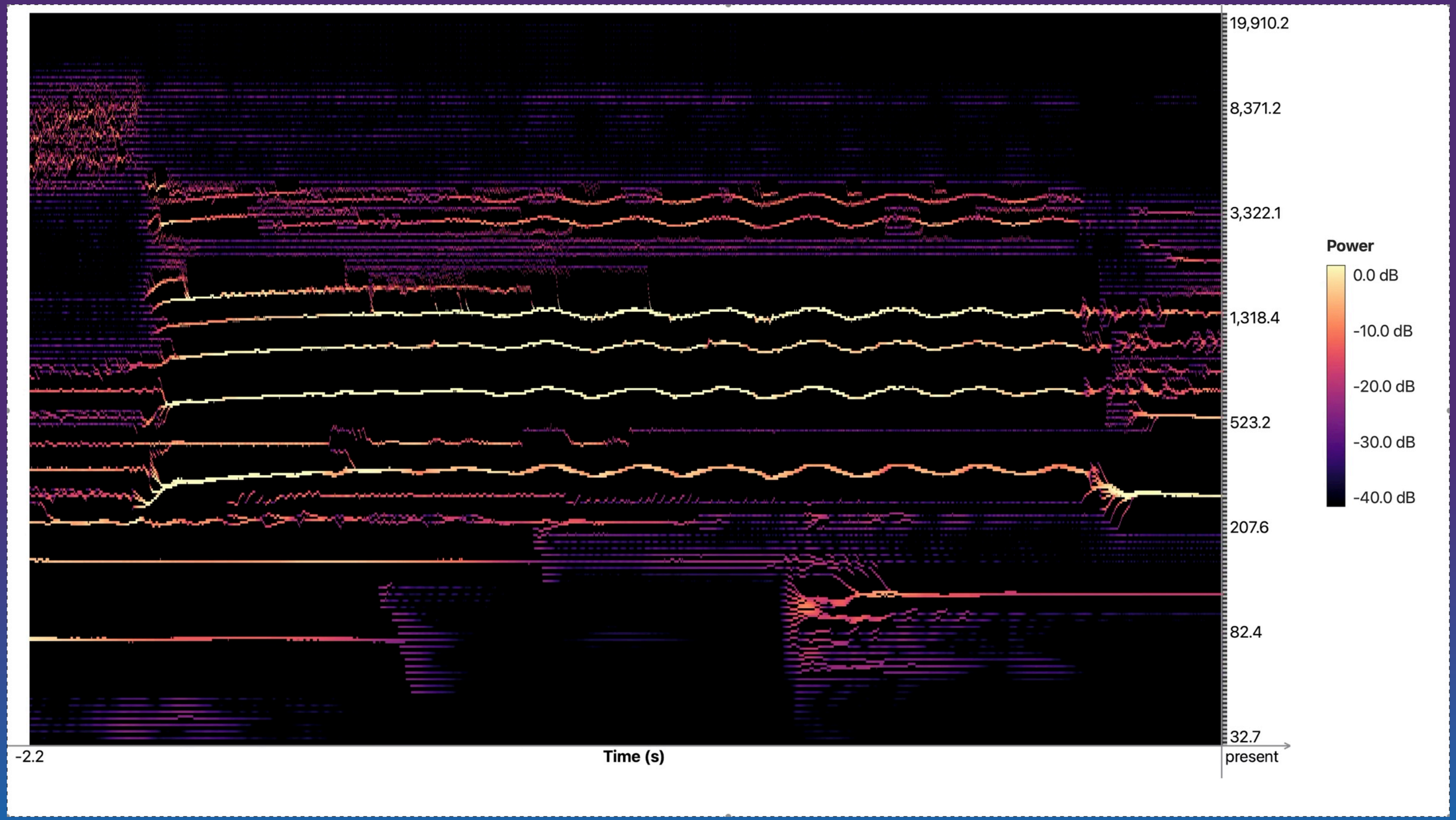
Low Latency, High Temporal and Frequency Resolution Spectral Analysis of Audio Signals

All About That Phase

Alexandre R.J. Francois



© 2026 Alexandre R. J. François

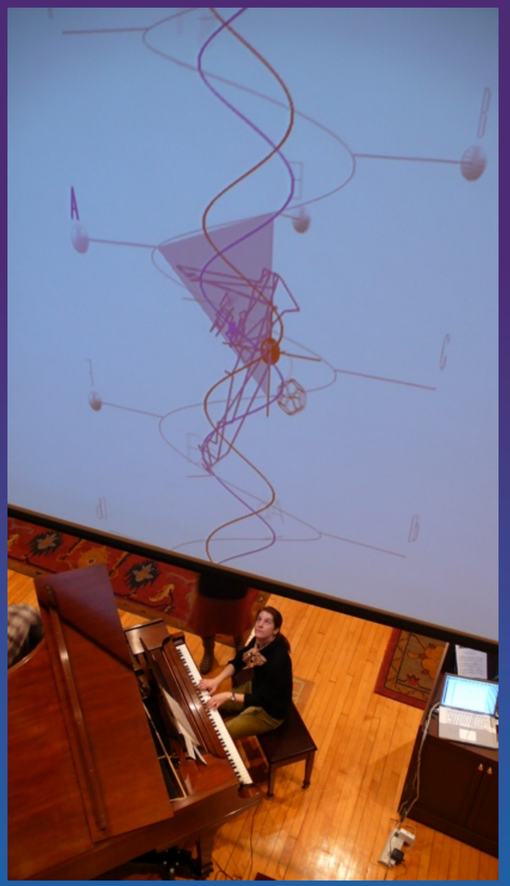


© 2026 Alexandre R. J. François

Alexandre R.J. François

- Computer scientist, software engineer, etc.
- Computer vision, interactive systems, computer graphics, **music computing**
- Academia, startups, larger companies
- Interactive systems, design, user experience (with computers)
  - For **brains**

MuSA.RT (2003+) with E. Chew



MIMI (2007+) with E. Chew



www.alexandrefrancois.org

© 2026 Alexandre R. J. François

Context and Motivation

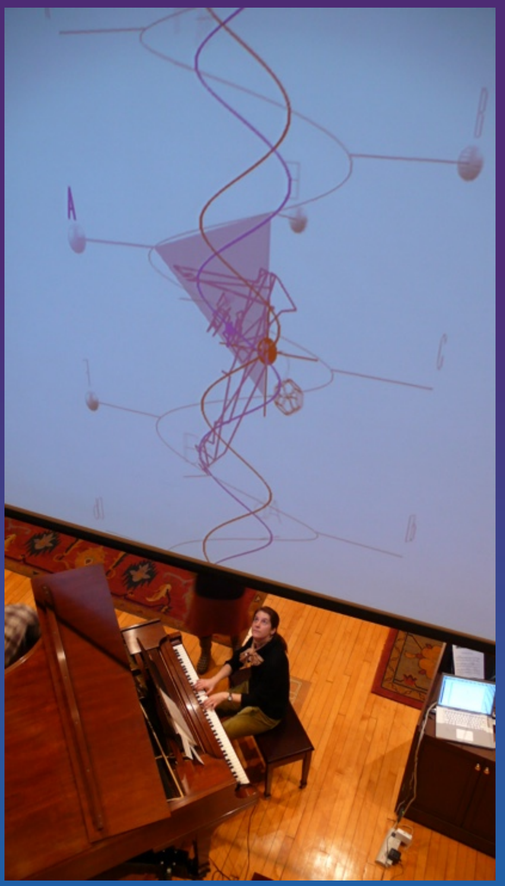
Which/how much of the frequencies of interest are present in the input signal?

- Input to interactive music analysis systems for visualization, improvisation, etc.
  - MIDI
- Perceptually relevant
- Low latency
- Computationally efficient

MIMI (2007+) with E. Chew



MuSA.RT (2003+) with E. Chew



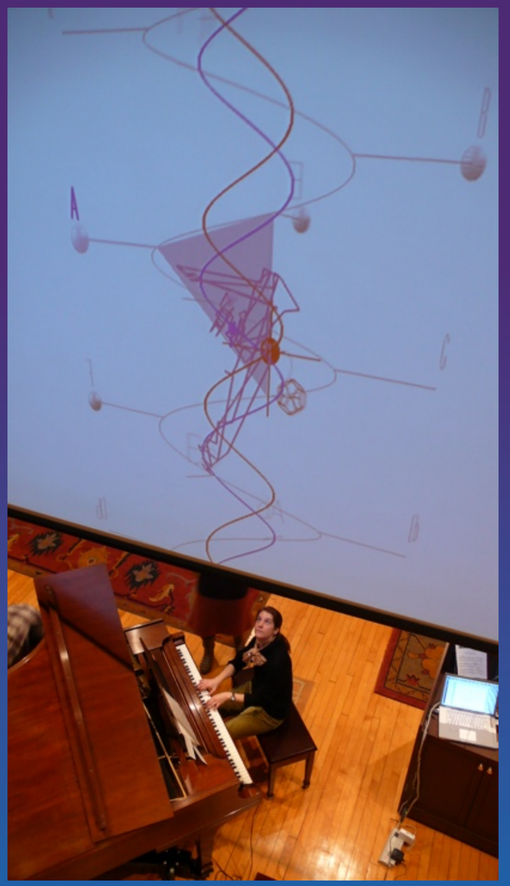
© 2026 Alexandre R. J. François

Context and Motivation

Which frequencies are present in the input signal (and how much)?

- Input to interactive music analysis systems for visualization, improvisation, etc.
  - MIDI
- Perceptually relevant
- Low latency
- Computationally efficient

MuSA.RT (2003+) with E. Chew



MIMI (2007+) with E. Chew



© 2026 Alexandre R. J. François

This Talk

- Goal: share what I have done, no matter how imperfect, so people can use it, improve on it, dream up and build applications aligned with their own interests…

- This is a personal project rooted in my past academic work, developed on week-ends, holidays and other breaks, over a number of years
  - Related open source material online since 2022
  - Does not represent the views of past or present employers

© 2026 Alexandre R. J. François

Resources

- *noFFT* open source module: python and C++ implementations, Jupyter notebooks with code used to generate the figures in the papers (and more…)
- Oscillators open source Swift package: reference implementations in Swift and C++
- **Oscillators app for iOS/iPadOS/MacOS (Apple Silicon, designed for iPad)**
- Resonate Youtube playlist

www.alexandrefrancois.org/Resonate



© 2026 Alexandre R. J. François

Outline

- Intro
- **Resonate model**
- Phase matters - instantaneous frequencies
- Self-tuning banks
- Frequency tracking spectrograms
- Synthesis
- Outro

© 2026 Alexandre R. J. François

Resonate

- Spectral analysis using a simple resonator model, assembled into banks
  - Efficient iterative formulation and implementation, no buffering
- Analogous to a bank of non-linear filters
- Not a better FFT - not trying to compute the DFT
- Not aware of this approach being used in a practical way for music/audio

- Reference publications:
  - Alexandre R.J. François, “Resonate: Efficient Low Latency Spectral Analysis of Audio Signals,” in Proceedings of the 50th Anniversary of the International Computer Music Conference 2025, pp. 251-258, Boston, MA, USA, 8-14 June 2025.
  - Alexandre R.J. François, “Real-Time, Low-Latency, High Resolution Audio Spectral Analysis: Phase Matters,” in Proceedings of the International Computer Music Conference 2026, Hamburg, Germany, 10-16 May 2026.

© 2026 Alexandre R. J. François

Resonator Model: Non-Linear Bandpass Filter

No buffering, iterative update, efficient implementation

- Each resonator has resonant frequency $f_k = \frac{\omega_k}{2\pi}$
- Heterodyning: iterative formulation of the exponentially weighted moving average (EWMA)

$$R_k(t) = (1 - \alpha_k)R_k(t - \Delta t) + \alpha_k x(t)e^{-i\omega_k t}$$

$$e^{-i\omega_k t} = P_k(t) = P_k(t - \Delta t)e^{-i\omega_k \Delta t}$$

$$R_k(t) = (1 - \alpha_k)R_k(t - \Delta t) + \alpha_k x(t)P_k(t - \Delta t)e^{-i\omega_k \Delta t}$$

- Lowpass filtering: $\tilde{R}_k(t) = (1 - \beta_k)\tilde{R}_k(t - \Delta t) + \beta_k R_k(t) \qquad \beta_k = \alpha_k$
- One single parameter $\alpha_k = 1 - e^{-\Delta t/\tau_k}$, relate to a time constant:* $\tau_k = -\Delta t/\ln(1 - \alpha_k)$
- Heuristic: time constant inversely proportional to frequency $\tau_f = log(1 + f)/f \qquad \alpha_f = 1 - e^{-\Delta t \frac{f}{log(1+f)}}$

© 2026 Alexandre R. J. François

*time for system response to a unit step function to reach ~62% of the original signal

Resonator Model: Non-Linear Bandpass Filter

No buffering, iterative update, efficient implementation

- Each resonator has resonant frequency $f_k = \frac{\omega_k}{2\pi}$
- Heterodyning: iterative formulation of the exponentially weighted moving average (EWMA)

$$R_k(t) = (1 - \alpha_k)R_k(t - \Delta t) +$$

$$R_k(t) = (1 - \alpha_k)R_k(t - \Delta t) +$$

- Lowpass filtering: $\tilde{R}_k(t) = (1 - \beta_k)\tilde{R}_k(t - \Delta t) +$
- One single parameter $\alpha_k = 1 - e^{-\Delta t/\tau_k}$, relate
- Heuristic: time constant inversely proportional to fr

```
// input signal X

// smoothed signal xx
// update at each sample x

xx = (1-alpha) * xx + alpha * x


// parameter alpha relates to time constant:
// time for system response to a unit step function
// to reach ~62% of the original signal

sampleDuration = 1.0 / sampleRate

tau = -sampleDuration / log(1.0 - alpha)
```

© 2026 Alexandre R. J. François

*time for system response to a unit step function to reach ~62% of the original signal

Resonator Model: Non-Linear Bandpass Filter

No buffering, iterative update, efficient implementation

- Each resonator has resonant frequency $f_k = \frac{\omega_k}{2\pi}$
- Heterodyning: iterative formulation of the exponentially weighted moving average (EWMA)

$$R_k(t) = (1 - \alpha_k)R_k(t - \Delta t) + \alpha_k x(t)e^{-i\omega_k t}$$

$$e^{-i\omega_k t} = P_k(t) = P_k(t - \Delta t)e^{-i\omega_k \Delta t}$$

$$R_k(t) = (1 - \alpha_k)R_k(t - \Delta t) + \alpha_k x(t)P_k(t - \Delta t)e^{-i\omega_k \Delta t}$$

- Lowpass filtering: $\tilde{R}_k(t) = (1 - \beta_k)\tilde{R}_k(t - \Delta t) + \beta_k R_k(t) \qquad \beta_k = \alpha_k$
- One single parameter $\alpha_k = 1 - e^{-\Delta t/\tau_k}$, relate to a time constant:* $\tau_k = -\Delta t/\ln(1 - \alpha_k)$
- Heuristic: time constant inversely proportional to frequency $\tau_f = log(1 + f)/f \qquad \alpha_f = 1 - e^{-\Delta t \frac{f}{log(1+f)}}$

© 2026 Alexandre R. J. François

*time for system response to a unit step function to reach ~62% of the original signal

Resonator Model: Non-Linear Bandpass Filter

No buffering, iterative update, efficient implementation

- Each resonator has resonant frequency $f_k = \frac{\omega_k}{2\pi}$
- Heterodyning: iterative formulation of the exponentially weighted moving average (EWMA)

$$e^{-i\omega_k t} = P_k(t) = P_k(t - \Delta t)e^{-i\omega_k \Delta t}$$

$$P_k(t - \Delta t)e^{-i\omega_k \Delta t}$$

```
class Phasor {
  var frequency: Float
  var sampleRate: Float

  // Phasor variables
  // Phasor: Z = Zc + j Zs
  // Multiplier: W = Wc + j Ws

  var Zc : Float = 1.0
  var Zs : Float = 0.0

  var Wc : Float = 0.0
  var Ws : Float = 0.0
  // pre-computed Wc + Ws
  var Wcps : Float = 0.0

  […]
}
```

- Lowpass filteri

$$t) \qquad \beta_k = \alpha_k$$

- One single par

ne constant:* $\tau_k = -\Delta t / \ln(1 - \alpha_k)$

- Heuristic: time

cy $\tau_f = log(1 + f)/f$ $\alpha_f = 1 - e^{-\Delta t \frac{f}{log(1+f)}}$

© 2026 Alexandre R. J. François

*time for system response to a unit step function to reach ~62% of the original signal

Resonator Model: Non-Linear Bandpass Filter

No buffering, iterative update, efficient implementation

- Each resonator has resonant frequency $f_k = \frac{\omega_k}{2\pi}$
- Heterodyning: iterative formulation of the exponentially weighted moving average (EWMA)

$$R_k(t) = (1 - \alpha_k)R_k(t - \Delta t) + \alpha_k x(t)e^{-i\omega_k t}$$

$$e^{-i\omega_k t} = P_k(t) = P_k(t - \Delta t)e^{-i\omega_k \Delta t}$$

$$R_k(t) = (1 - \alpha_k)R_k(t - \Delta t) + \alpha_k x(t)P_k(t - \Delta t)e^{-i\omega_k \Delta t}$$

- Lowpass filtering: $\tilde{R}_k(t) = (1 - \beta_k)\tilde{R}_k(t - \Delta t) + \beta_k R_k(t) \qquad \beta_k = \alpha_k$
- One single parameter $\alpha_k = 1 - e^{-\Delta t/\tau_k}$, relate to a time constant:* $\tau_k = -\Delta t/\ln(1 - \alpha_k)$
- Heuristic: time constant inversely proportional to frequency $\tau_f = log(1 + f)/f \qquad \alpha_f = 1 - e^{-\Delta t \frac{f}{log(1+f)}}$

© 2026 Alexandre R. J. François

*time for system response to a unit step function to reach ~62% of the original signal

Resonator Model: Non-Linear

No buffering, iterative update, efficient imp

- Each resonator has resonant frequency $f_k = \frac{\omega_k}{2\pi}$
- Heterodyning: iterative formulation of the exponentially weighted moving a

$$R_k(t) = (1 - \alpha_k)R_k(t - \Delta t) + \alpha_k x(t)e^{-i\omega_k t}$$

$$e^{-i\omega_k t} = P_k(t) =$$

$$R_k(t) = (1 - \alpha_k)R_k(t - \Delta t) + \alpha_k x(t)P_k(t - \Delta t)e^{-i\omega_k \Delta t}$$

- Lowpass filtering: $\tilde{R}_k(t) = (1 - \beta_k)\tilde{R}_k(t - \Delta t) + \beta_k R_k(t) \qquad \beta_k = \alpha_k$
- One single parameter $\alpha_k = 1 - e^{-\Delta t/\tau_k}$, relate to a time constant:* $\tau_k =$
- Heuristic: time constant inversely proportional to frequency $\tau_f = log(1 -$

```
class Resonator : Phasor {
  // complex: r = c + j s
  var c: Float = 0.0
  var s: Float = 0.0

  // Smoothed resonator output
  var cc: Float = 0.0
  var ss: Float = 0.0

  […]
}

// Update with sample
alphaSample = alpha * sample
omAlpha = 1.0 - alpha
c = omAlpha * c + alphaSample * Zc
s = omAlpha * s + alphaSample * Zs


// Compute next value of the phasor
// Z <- Z * W
// complex multiplication
// with 3 real multiplications

ac = Wc*Zc
bd = Ws*Zs
abcd = (Wcps) * (Zc+Zs)
Zc = ac - bd
Zs = abcd - ac - bd
```

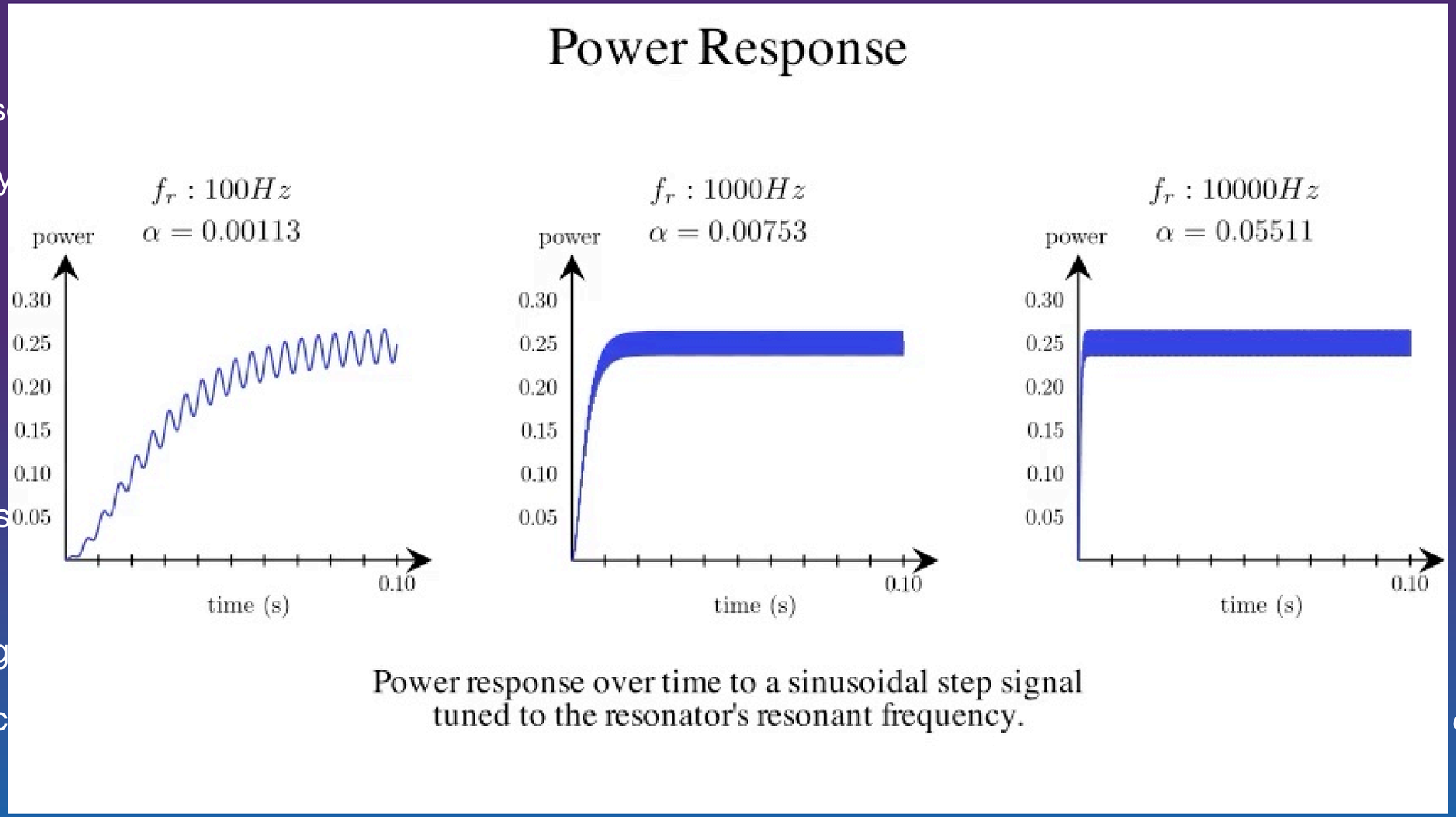
© 2026 Alexandre R. J. François

*time for system response to a unit step function to reach ~62% of the original signal

Resonator Model: Non-Linear Bandpass Filter

No buffering, iterative update, efficient implementation

- Each res
- Heterody
- Lowpass
- One sing
- Heuristic



Power response over time to a sinusoidal step signal tuned to the resonator's resonant frequency.

$$e^{-\Delta t \frac{f}{log(1+f)}}$$

*time for system response to a unit step function to reach ~62% of the original signal

© 2026 Alexandre R. J. François

Resonator Model: Non-Linear Bandpass Filter

No buffering, iterative update, efficient implementation

- Each resonator has resonant frequency $f_k = \frac{\omega_k}{2\pi}$
- Heterodyning: iterative formulation of the exponentially weighted moving avera

$$R_k(t) = (1 - \alpha_k)R_k(t - \Delta t) + \alpha_k x(t)e^{-i\omega_k t}$$

$$e^{-i\omega_k t} = P_k(t) = P_k$$

$$R_k(t) = (1 - \alpha_k)R_k(t - \Delta t) + \alpha_k x(t)P_k(t - \Delta t)e^{-i\omega_k \Delta t}$$

- Lowpass filtering:

$$\tilde{R}_k(t) = (1 - \beta_k)\tilde{R}_k(t - \Delta t) + \beta_k R_k(t) \qquad \beta_k = \alpha_k$$

```
class Resonator : Phasor {
// complex: r = c + j s
var c: Float = 0.0
var s: Float = 0.0

// Smoothed resonator
output
var cc: Float = 0.0
var ss: Float = 0.0

[…]
}

omBeta = 1.0 - beta
cc = omBeta * cc + beta * c
ss = omBeta * ss + beta * s
```

- One single parameter $\alpha_k = 1 - e^{-\Delta t/\tau_k}$, relate to a time constant:* $\tau_k = -\Delta t/\ln(1 - \alpha_k)$
- Heuristic: time constant inversely proportional to frequency $\tau_f = log(1 + f)/f$ $\alpha_f = 1 - e^{-\Delta t \frac{f}{log(1+f)}}$

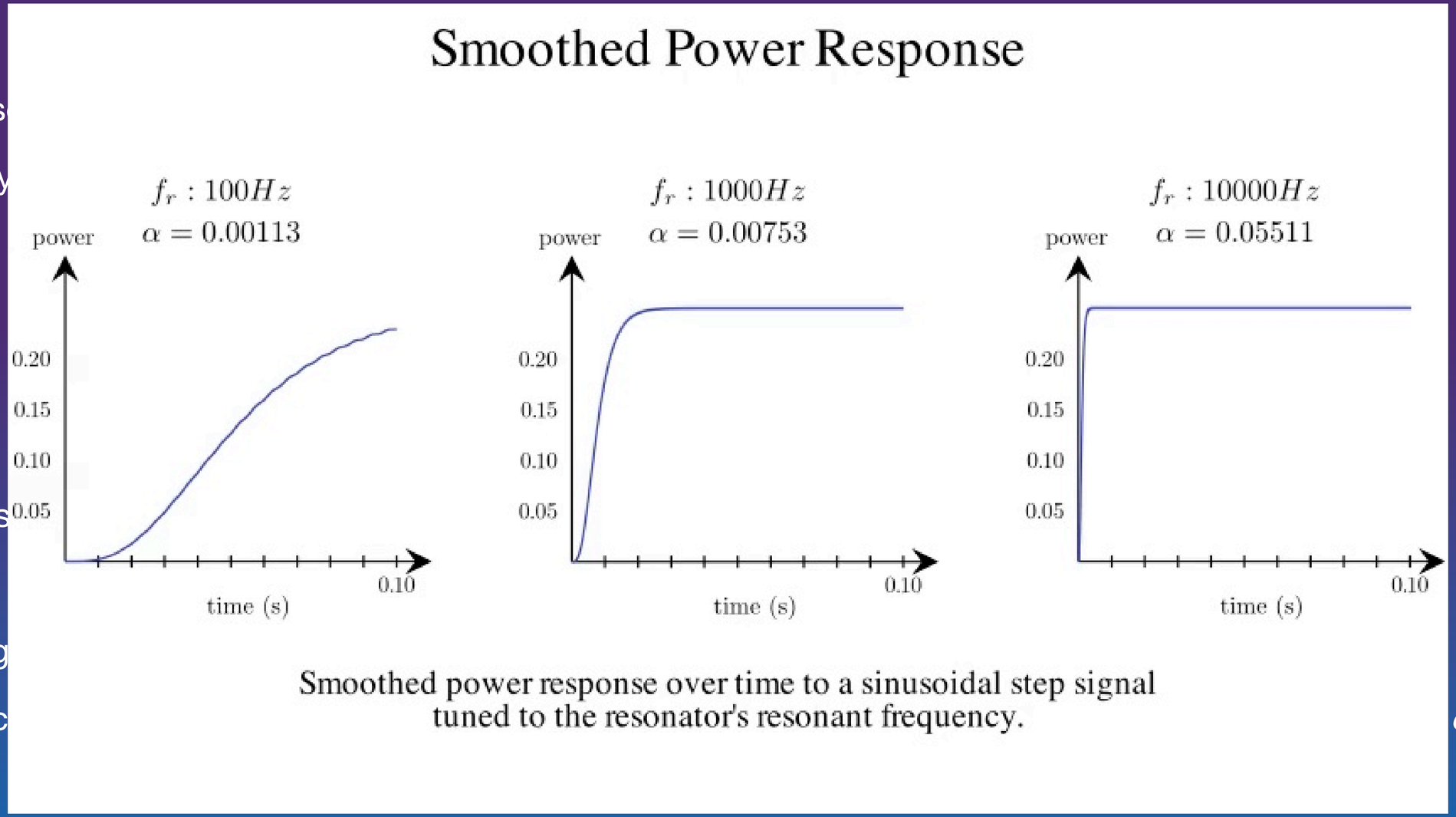
© 2026 Alexandre R. J. François

*time for system response to a unit step function to reach ~62% of the original signal

Resonator Model: Non-Linear Bandpass Filter

No buffering, iterative update, efficient implementation

- Each res
- Heterody
- Lowpass
- One sing
- Heuristic



Smoothed power response over time to a sinusoidal step signal tuned to the resonator's resonant frequency.

$$e^{-\Delta t \frac{f}{log(1+f)}}$$

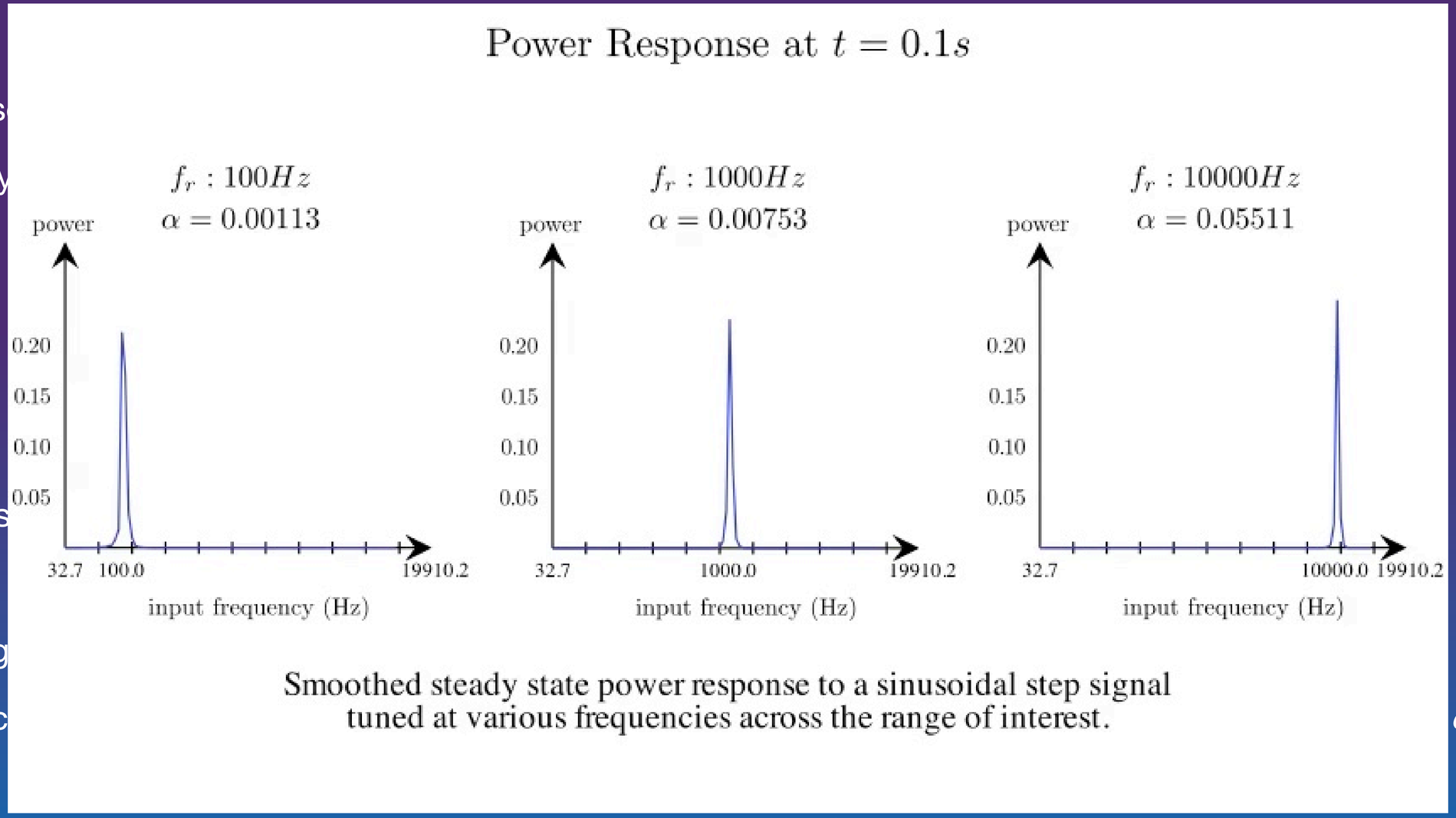
*time for system response to a unit step function to reach ~62% of the original signal

© 2026 Alexandre R. J. François

Resonator Model: Non-Linear Bandpass Filter

No buffering, iterative update, efficient implementation

- Each res
- Heterody
- Lowpass
- One sing
- Heuristic



Smoothed steady state power response to a sinusoidal step signal tuned at various frequencies across the range of interest.

$$e^{-\Delta t \frac{f}{log(1+f)}}$$

*time for system response to a unit step function to reach ~62% of the original signal

© 2026 Alexandre R. J. François

Computational Cost

Per resonator

- Memory: <10 Floats
- Computations **per sample** update:
  - a handful of multiplications and additions
  - Some occasional overhead to compensate for numerical approximations
  - Complexity is constant
- Overall complexity is linear in the number of samples processed
- Works across whole frequency range with same efficiency

© 2026 Alexandre R. J. François

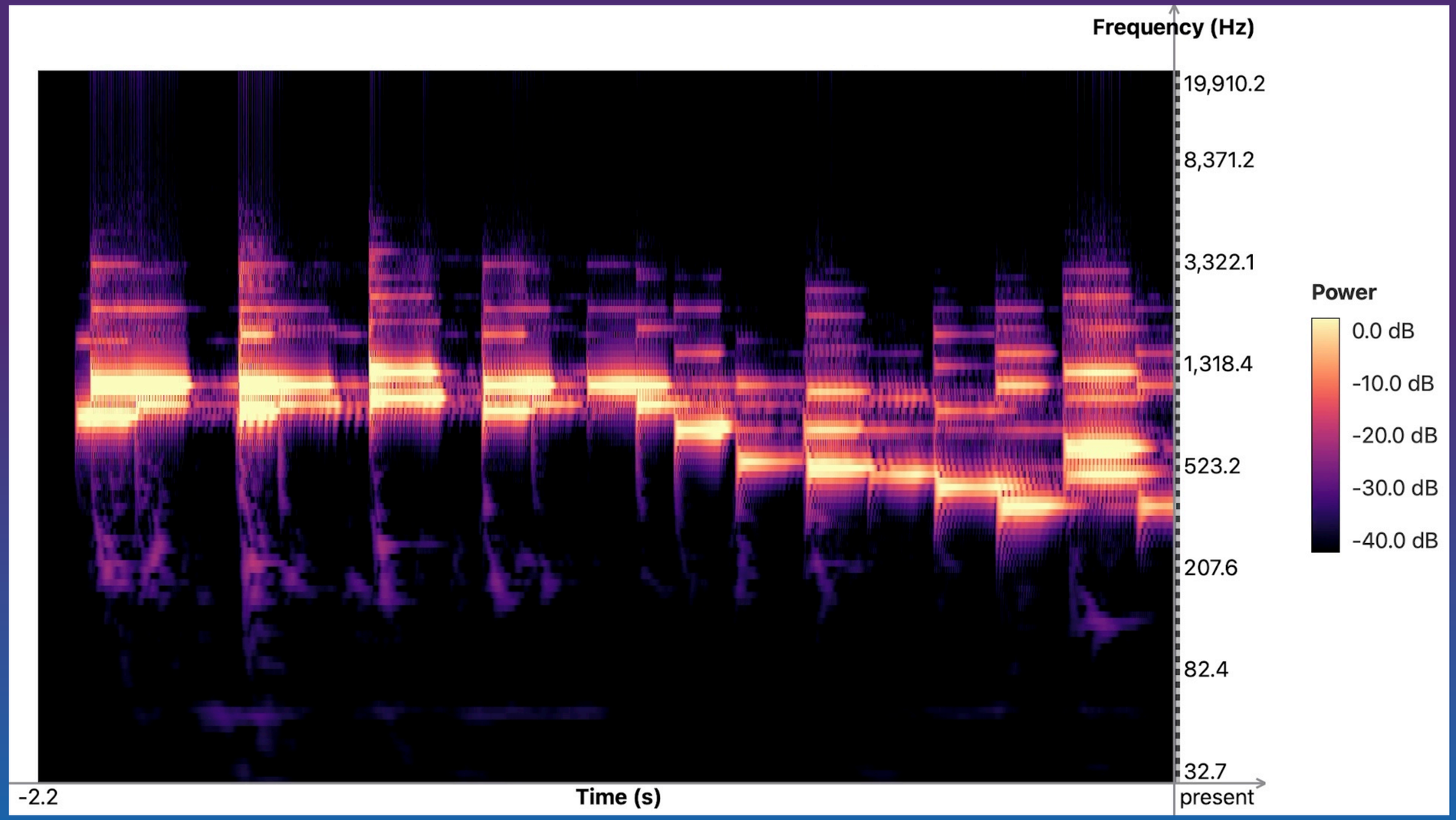
Resonator Banks

- Arbitrary number of resonators
- Directly tuned at frequencies of interest, arbitrarily distributed
  - Log frequency scale (like CQT), mel scale, etc.
  - Dynamics
- Resonators are independent
  - Parallel processing of each sample
  - With SIMD acceleration, updating a 112 resonator bank takes less than 120 ns per sample (iPhone/iPad - Accelerate Framework)

© 2026 Alexandre R. J. François

Spectrogram

Who’s Loving You - The Jacksons 5



© 2026 Alexandre R. J. François

Spectrogram

- High temporal resolution and low latency
  - one measurement per input sample, including phase
- Frequency resolution is not great (“leakage”)
  - STFT based methods apply window shaping to mitigate leakage
- What does the power response of each resonator represent?
  - Proportional to:
    - Amplitude of input signal component
    - Distance between resonant frequency and actual component frequency

© 2026 Alexandre R. J. François

Outline

- Intro
- Resonate model
- **Phase matters - instantaneous frequencies**
- Self-tuning banks
- Frequency tracking spectrograms
- Synthesis
- Outro

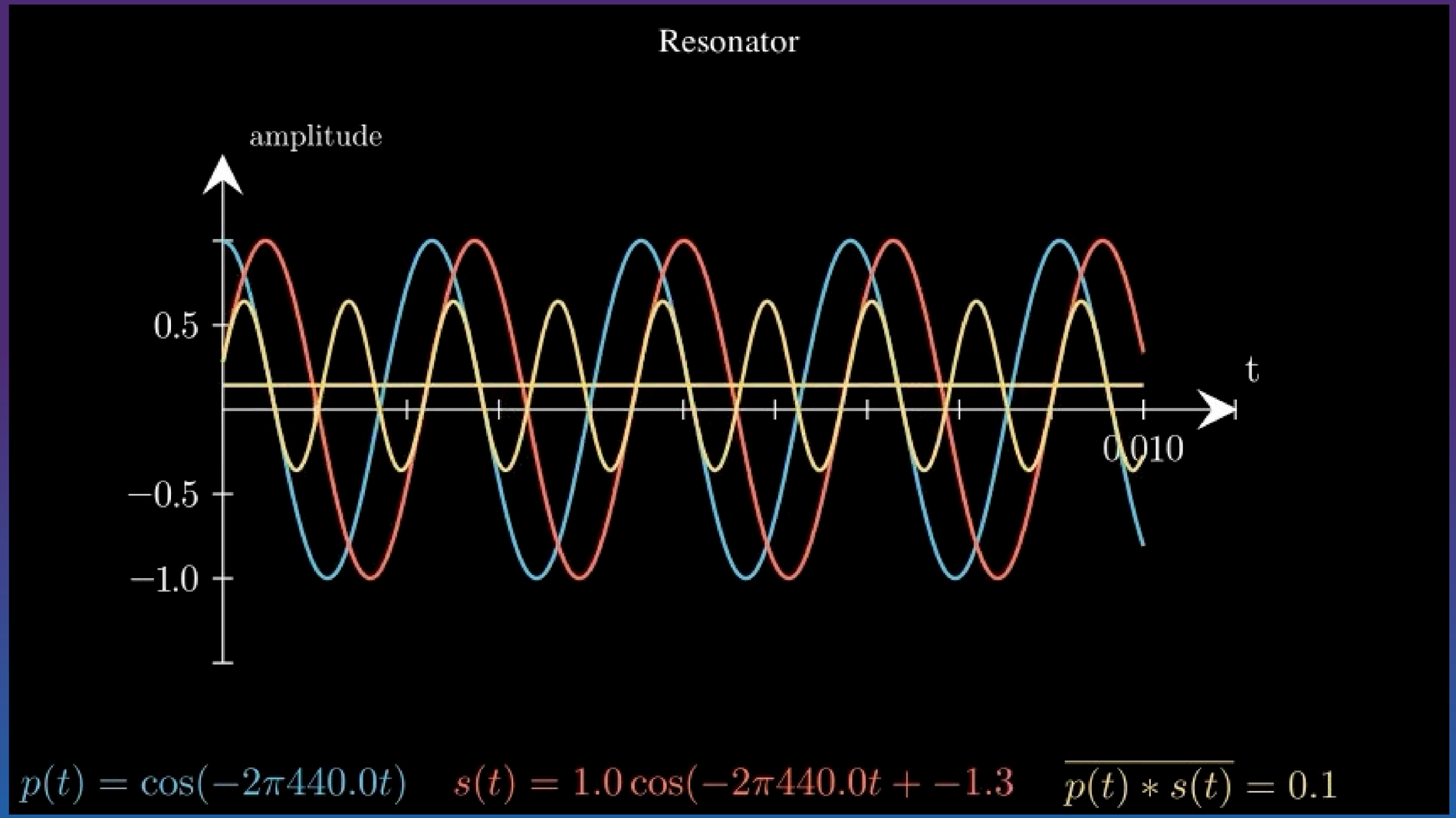
© 2026 Alexandre R. J. François

Phase Vocoder

- An analysis-synthesis method to manipulate audio signal in time and frequency
  - Phase information is crucial for (re-)synthesis
- Phase vocoder papers I wish I had read/understood better earlier…
  - Portnoff,1976 - FFT implementation
  - Moorer, 1978 - use in music
  - Dolson, 1986 - “The Phase Vocoder: A Tutorial”
    - Filterbank and FFT interpretations
- Principles to start from
  - Build on instantaneous frequencies estimation
  - Avoid significant mathematical and computational complexity introduced by the use of use STFT and iSTFT: windowing, phase locking, etc.

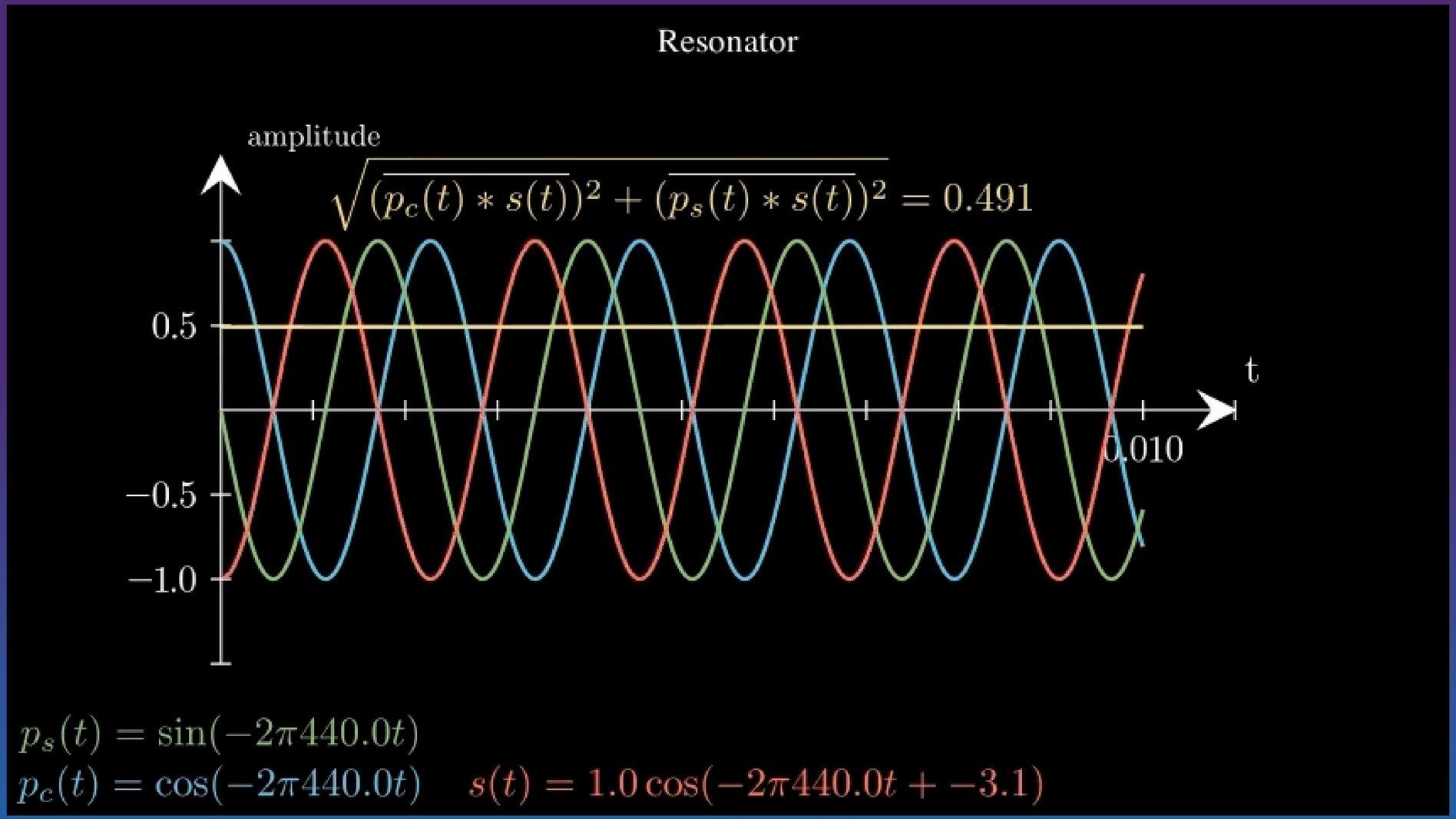
© 2026 Alexandre R. J. François

Phase - Single Sine



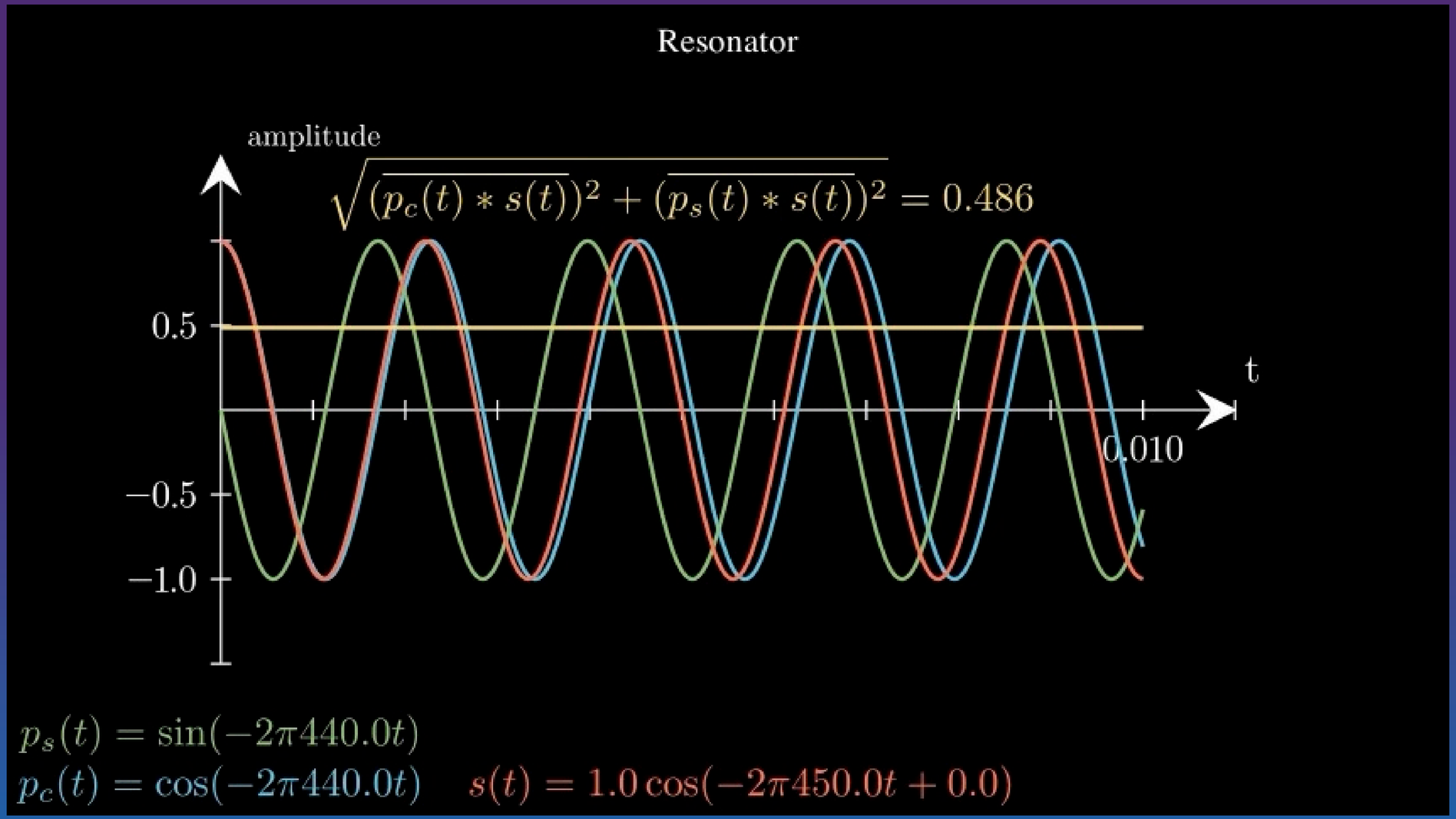
© 2026 Alexandre R. J. François

Phase - Complex



© 2026 Alexandre R. J. François

Near Resonant Frequency



© 2026 Alexandre R. J. François

Instantaneous Frequency

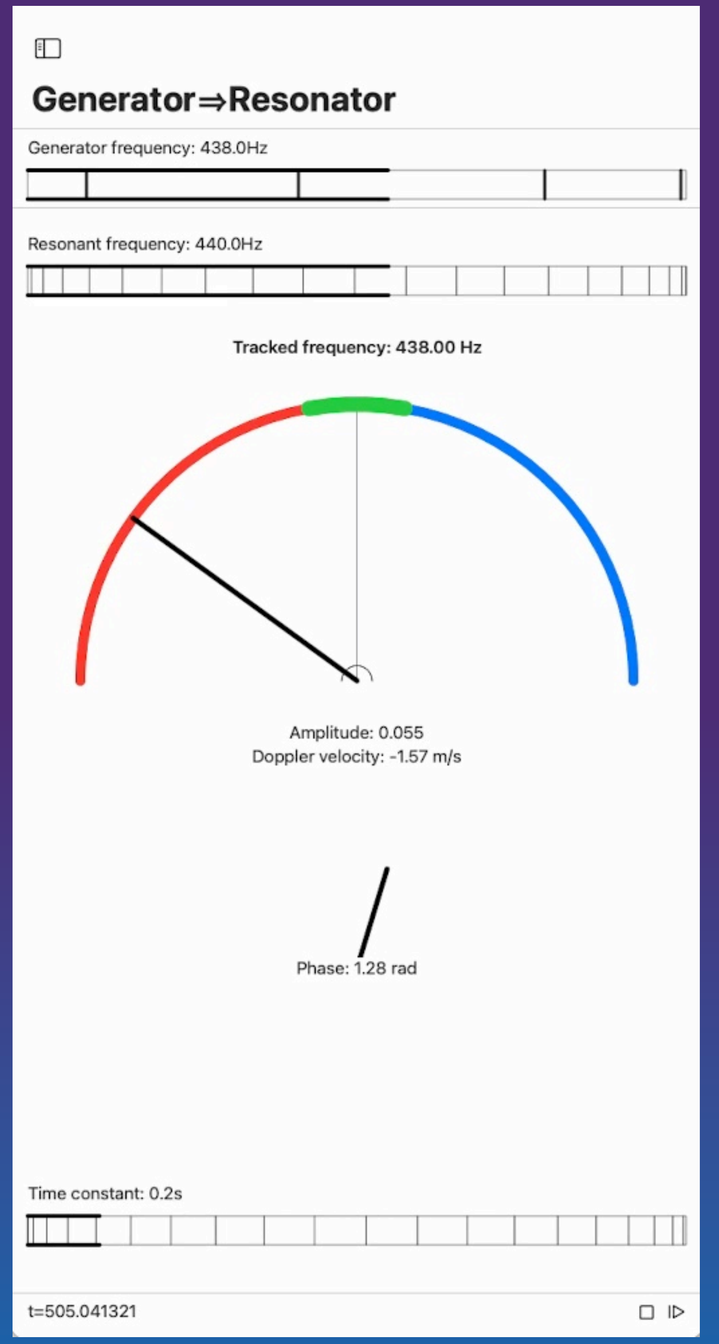
- Estimate from phase difference
  - for resonators whose amplitude is above a certain threshold
- Calculated efficiently, at each sample
  - no phase unwrapping concern
  - Amplitude proportional to distance to resonant frequency

$$f_k^i(t) = f_k^r + \frac{\Delta\phi_k(t)}{2\pi\Delta t}$$

$$D_k(t) = \tilde{R}_k(t)\overline{\tilde{R}_k}(t - \Delta t)$$

$$\Delta\phi_k(t) = Arg(D_k(t))$$

$$\tilde{f_k^i}(t) = f_k^i(t - \Delta t) + \gamma_k \frac{\Delta\phi_k(t)}{2\pi\Delta t} \qquad \gamma_k = \alpha_k/2$$



© 2026 Alexandre R. J. François

Instantaneous Frequency

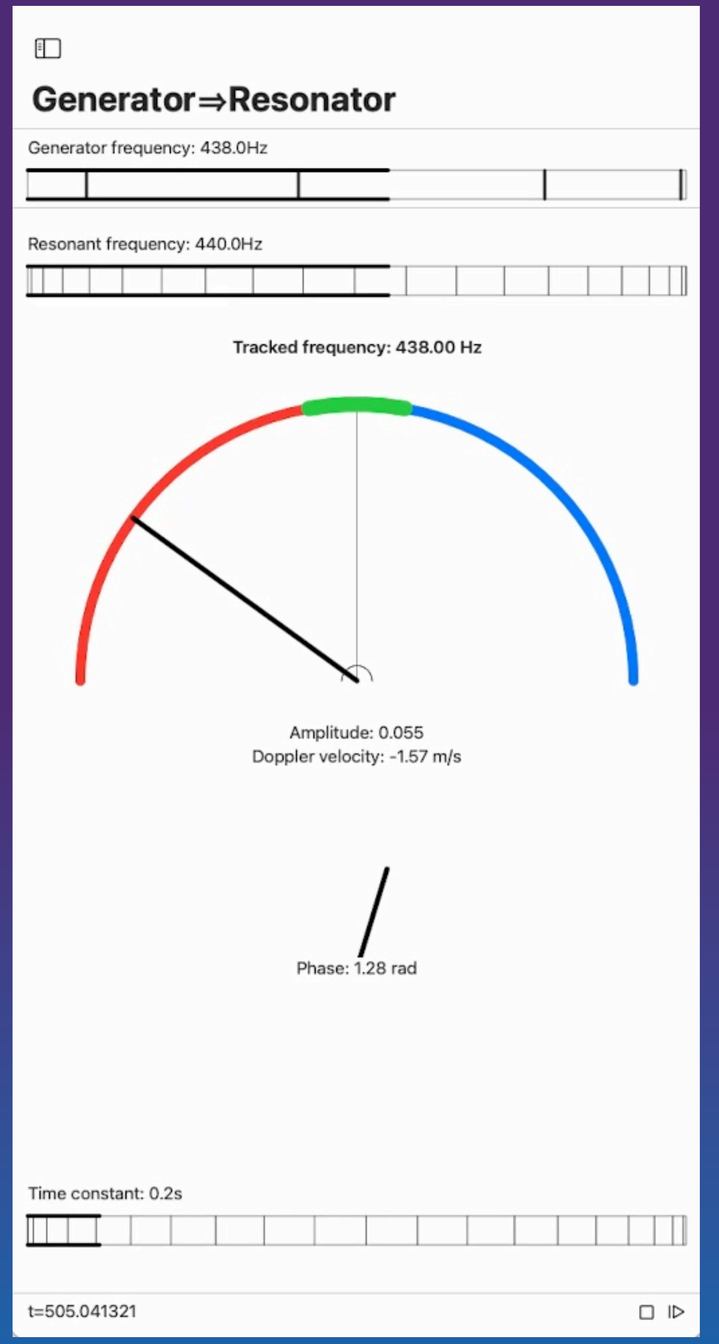
- Estimate from phase difference
  - for resonators whose amplitude is above a certain threshold
- Calculated efficiently, at each sample
  - no phase unwrapping concern
  - Amplitude proportional to distance to resonant frequency

$$f_k^i(t) = f_k^r + \frac{\Delta\phi_k(t)}{2\pi\Delta t}$$

$$D_k(t) = \tilde{R}_k(t)\overline{\tilde{R}_k}(t - \Delta t)$$

$$\Delta\phi_k(t) = Arg(D_k(t))$$

$$\tilde{f}_k^i(t) = f_k^i(t - \Delta t) + \gamma_k \frac{\Delta\phi_k(t)}{2\pi\Delta t} \qquad \gamma_k = \alpha_k/2$$



© 2026 Alexandre R. J. François

Instantaneous Frequency

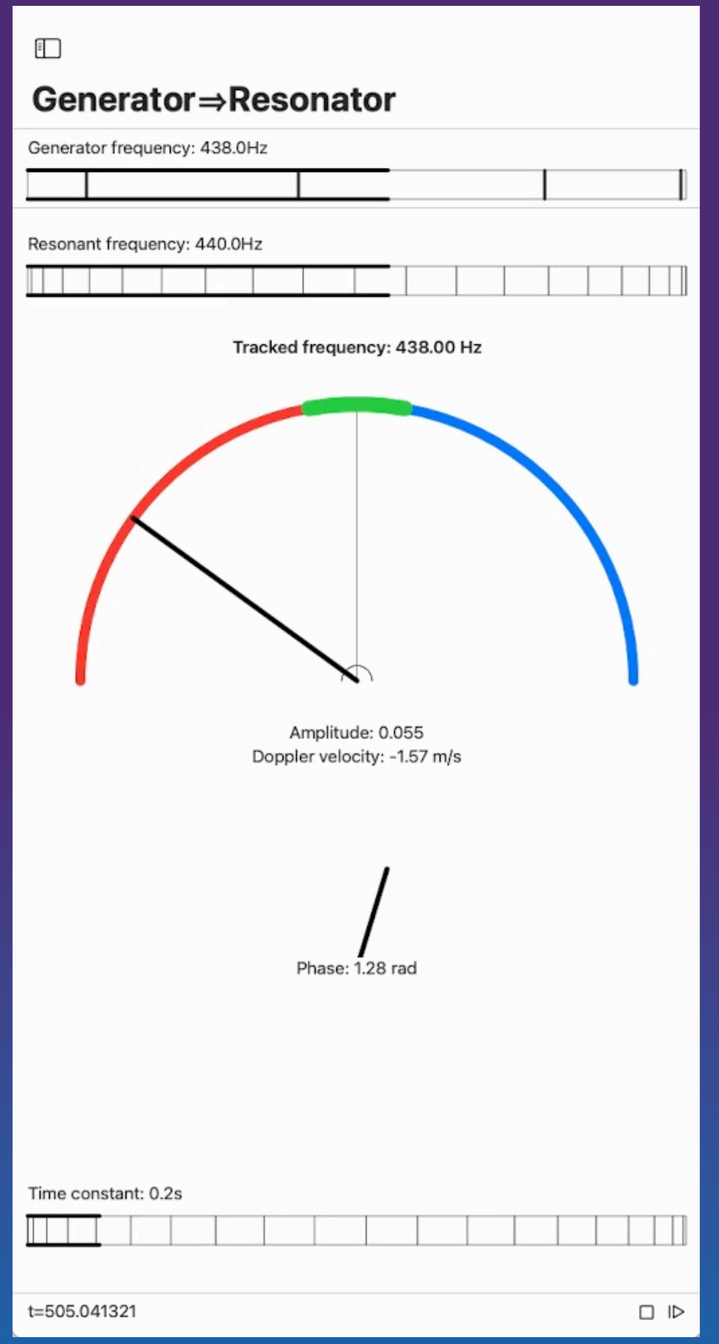
- Estimate from phase difference
  - for resonators whose amplitude is above a certain threshold
- Calculated efficiently, at each sample
  - no phase unwrapping concern
  - Amplitude proportional to distance to resonant frequency

$$f_k^i(t) = f_k^r + \frac{\Delta\phi_k(t)}{2\pi\Delta t}$$

$$D_k(t) = \tilde{R}_k(t)\overline{\tilde{R}_k}(t-\Delta t)$$

$$\Delta\phi_k(t) = Arg(D_k(t))$$

$$\tilde{f}_k^i(t) = f_k^i(t-\Delta t) + \gamma_k \frac{\Delta\phi_k(t)}{2\pi\Delta t} \qquad \gamma_k = \alpha_k/2$$



© 2026 Alexandre R. J. François

Instantaneous Frequency

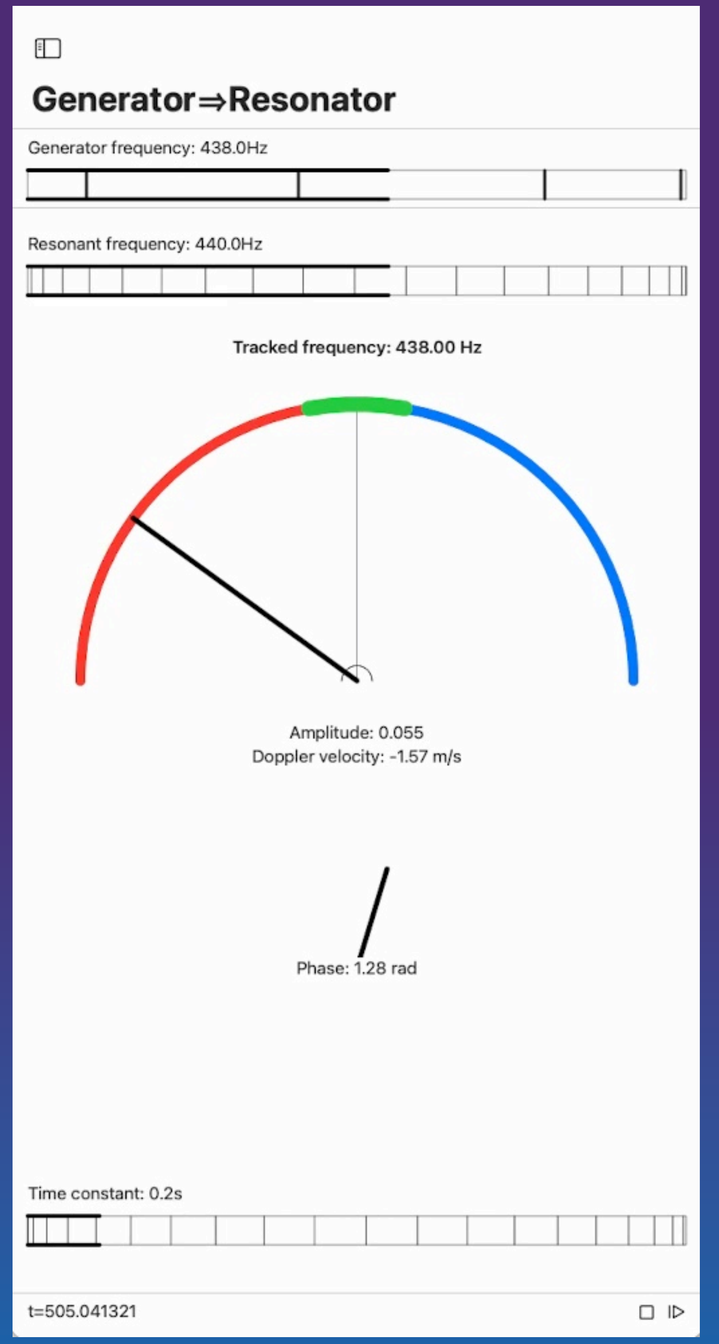
- Estimate from phase difference
  - for resonators whose amplitude is above a certain threshold
- Calculated efficiently, at each sample
  - no phase unwrapping concern
  - Amplitude proportional to distance to resonant frequency

$$f_k^i(t) = f_k^r + \frac{\Delta\phi_k(t)}{2\pi\Delta t}$$

$$D_k(t) = \tilde{R}_k(t)\overline{\tilde{R}_k}(t - \Delta t)$$

$$\Delta\phi_k(t) = Arg(D_k(t))$$

$$\tilde{f}_k^i(t) = f_k^i(t - \Delta t) + \gamma_k \frac{\Delta\phi_k(t)}{2\pi\Delta t} \qquad \gamma_k = \alpha_k/2$$



© 2026 Alexandre R. J. François

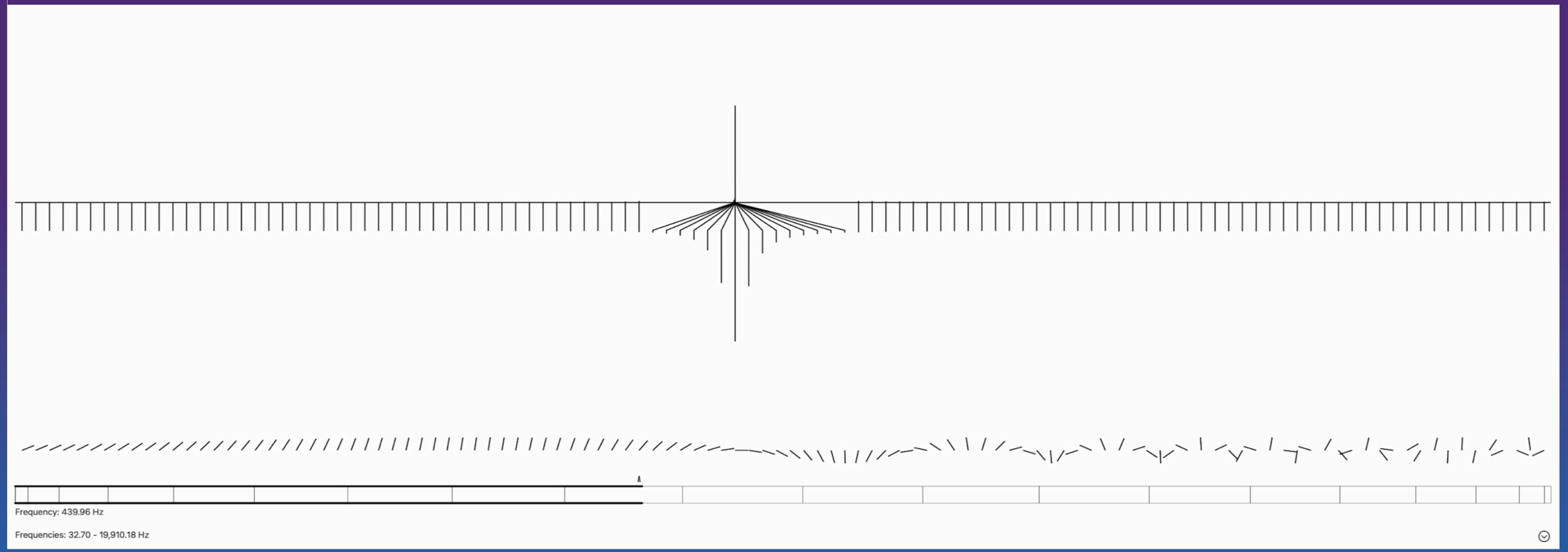
Instantaneous Frequencies

Amplitude proportional to distance to resonant frequency

f^i

f^r

$\Delta\phi$



© 2026 Alexandre R. J. François

Outline

- Intro
- Resonate model
- Phase matters - instantaneous frequencies
- **Self-tuning banks**
- Frequency tracking spectrograms
- Synthesis
- Outro

© 2026 Alexandre R. J. François

Self-Tuning Bank

Frequency tracking

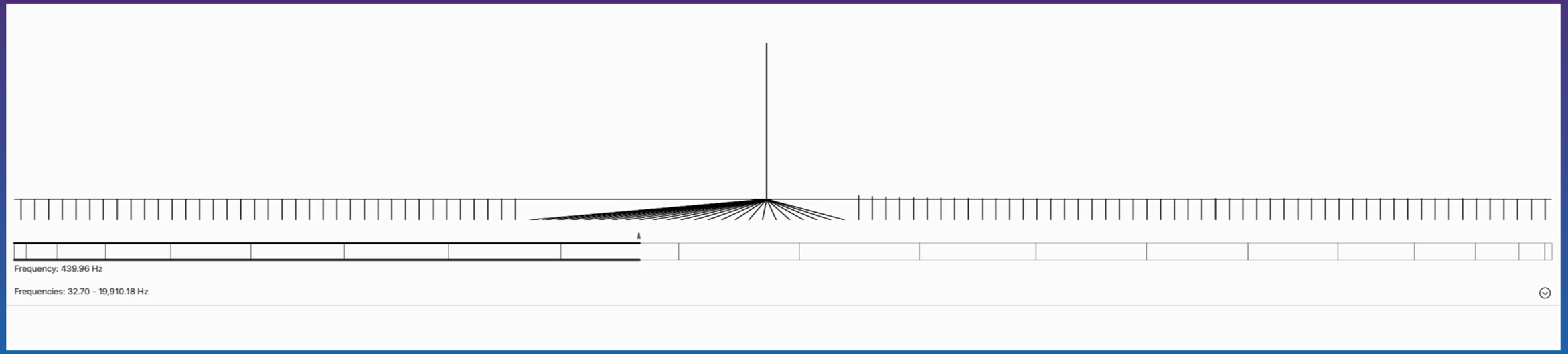
- Resonators have natural frequency which is resting resonant frequency

$$f_k^n = \frac{\omega_k^n}{2\pi}$$

- When magnitude is above a certain threshold,
  resonant frequency tracks estimated instantaneous frequency

$$\omega_k(t) = \omega_k(t - \Delta t) + \gamma_k \frac{\Delta\phi_k(t)}{\Delta t} \qquad \gamma_k = \alpha_k/2$$

$$R_k(t) = (1 - \alpha_k)R_k(t - \Delta t) + \alpha_k x(t) P_k(t - \Delta t) e^{-i\omega_k(t)\Delta t}$$



© 2026 Alexandre R. J. François

Self-Tuning Bank

Frequency tracking

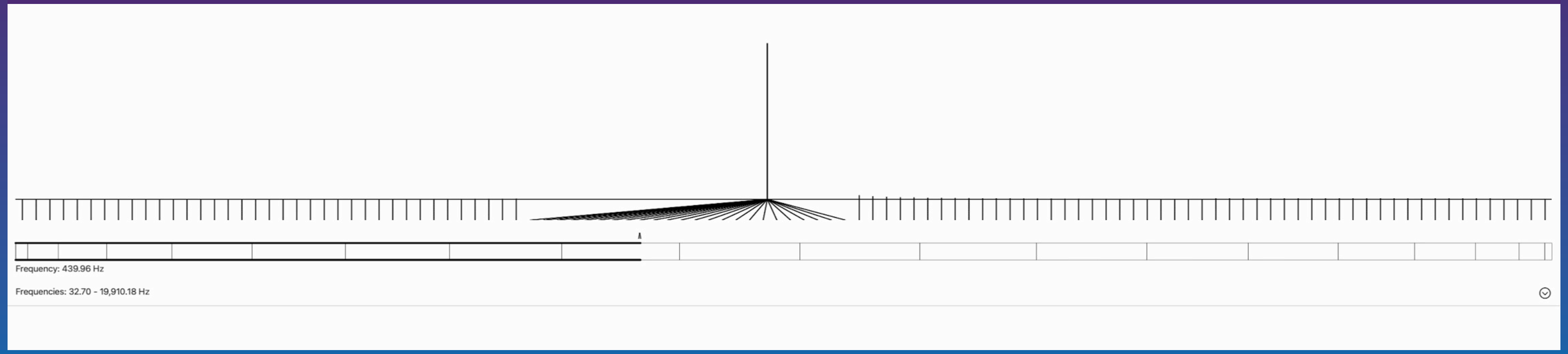
- Resonators have natural frequency which is resting resonant frequency

$$f_k^n = \frac{\omega_k^n}{2\pi}$$

- When magnitude is above a certain threshold,
  resonant frequency tracks estimated instantaneous frequency

$$\omega_k(t) = \omega_k(t - \Delta t) + \gamma_k \frac{\Delta\phi_k(t)}{\Delta t} \qquad \gamma_k = \alpha_k/2$$

$$R_k(t) = (1 - \alpha_k)R_k(t - \Delta t) + \alpha_k x(t) P_k(t - \Delta t) e^{-i\omega_k(t)\Delta t}$$



© 2026 Alexandre R. J. François

Self-Tuning Bank

Frequency tracking

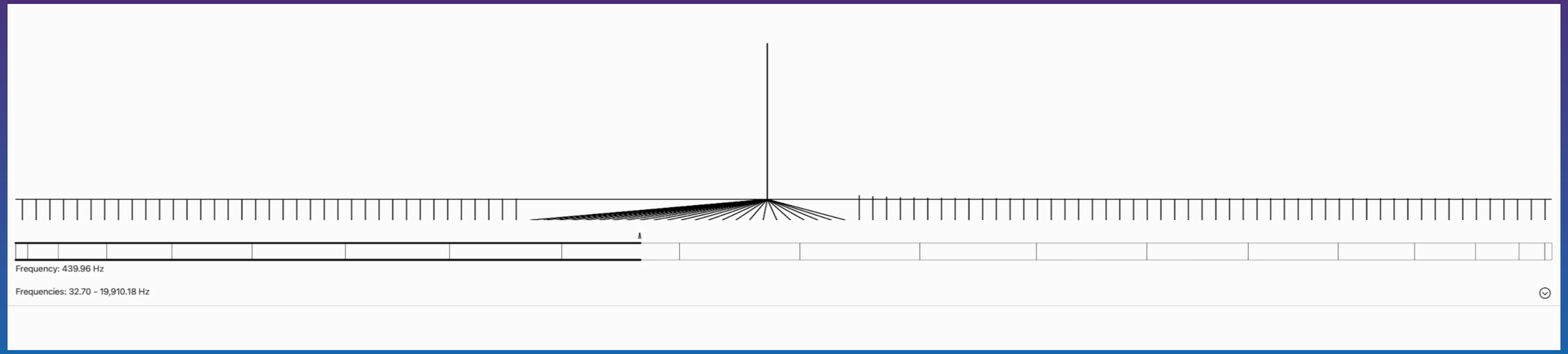
- Resonators have natural frequency which is resting resonant frequency

$$f_k^n = \frac{\omega_k^n}{2\pi}$$

- When magnitude is above a certain threshold,
  resonant frequency tracks estimated instantaneous frequency

$$\omega_k(t) = \omega_k(t - \Delta t) + \gamma_k \frac{\Delta\phi_k(t)}{\Delta t} \qquad \gamma_k = \alpha_k/2$$

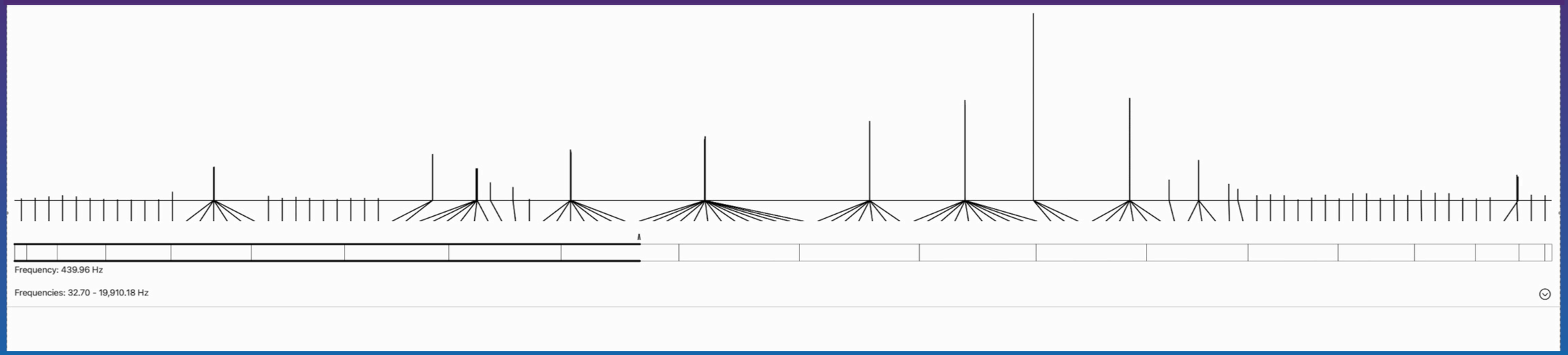
$$R_k(t) = (1 - \alpha_k)R_k(t - \Delta t) + \alpha_k x(t) P_k(t - \Delta t)e^{-i\omega_k(t)\Delta t}$$



© 2026 Alexandre R. J. François

Self-Tuning Bank

Who’s Loving You - The Jacksons 5



© 2026 Alexandre R. J. François

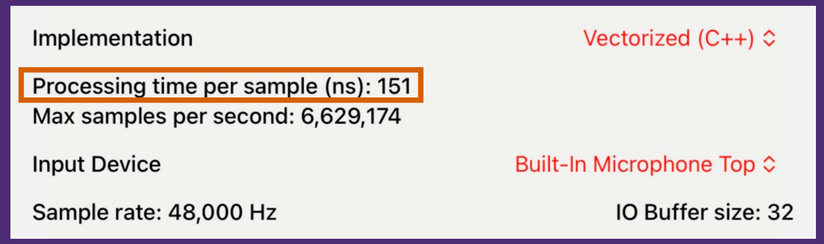
Implementation

- Frequency tracking only adds a few more operations per sample computation
- Arctan for explicit frequency/angular velocity computation
  - Not too bad: hardware support
- Approximate phase locking methods keep frequency implicit
  - Similar behavior and dynamics, no transcendental functions
- With SIMD acceleration, updating a 112 resonator bank takes well under 1µs per sample (1 sample duration at 48kHz is about 21µs)

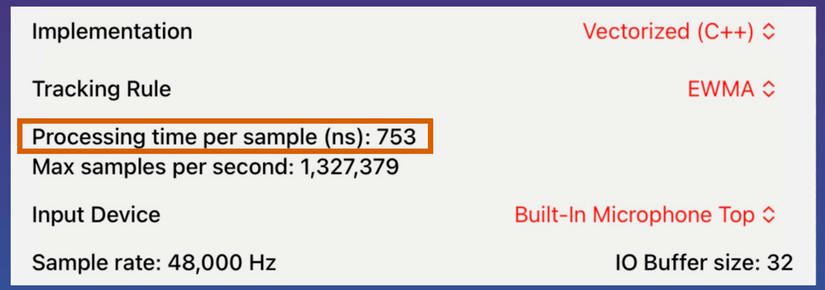
© 2026 Alexandre R. J. François

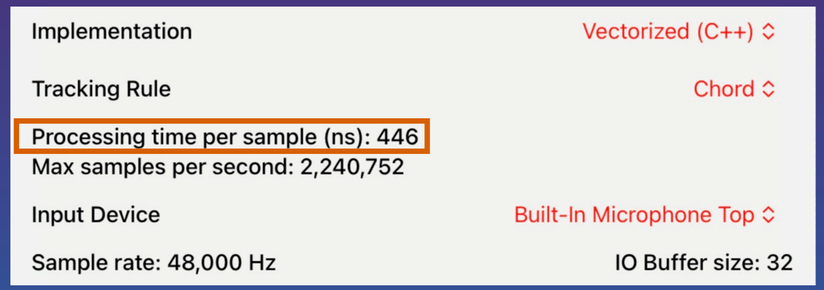
Performance (SIMD - C++)

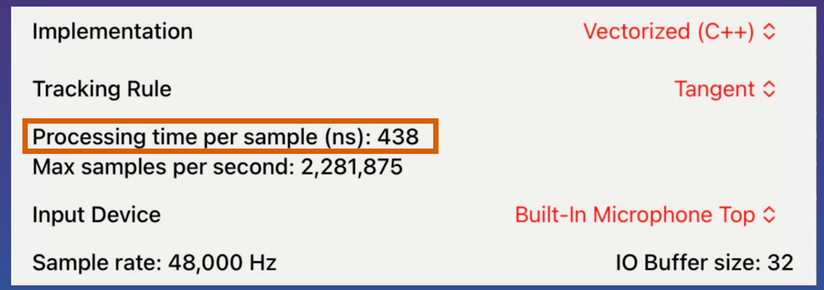
Non-tracking



Tracking







Resonate processing times per sample on iPad A16 - screenshots from Spectrogram and Tracking Spectrogram settings screen

© 2026 Alexandre R. J. François

Output and properties

- The resonators amplitudes are directly proportional to the matching component's strength in the signal (after convergence)
- If the density and distribution of natural frequencies is adequate, the actual tuning is irrelevant - the bank is "self-tuning"
  - The resonant frequencies track the frequencies that are actually present in the input signal
  - Nearby resonators converge towards the **locally dominant component** frequency
  - Density and distribution of natural frequencies are application dependent and could be learned
- Select the resonators whose tracked frequency is closest to their natural frequency

$$T(t)$$

© 2026 Alexandre R. J. François

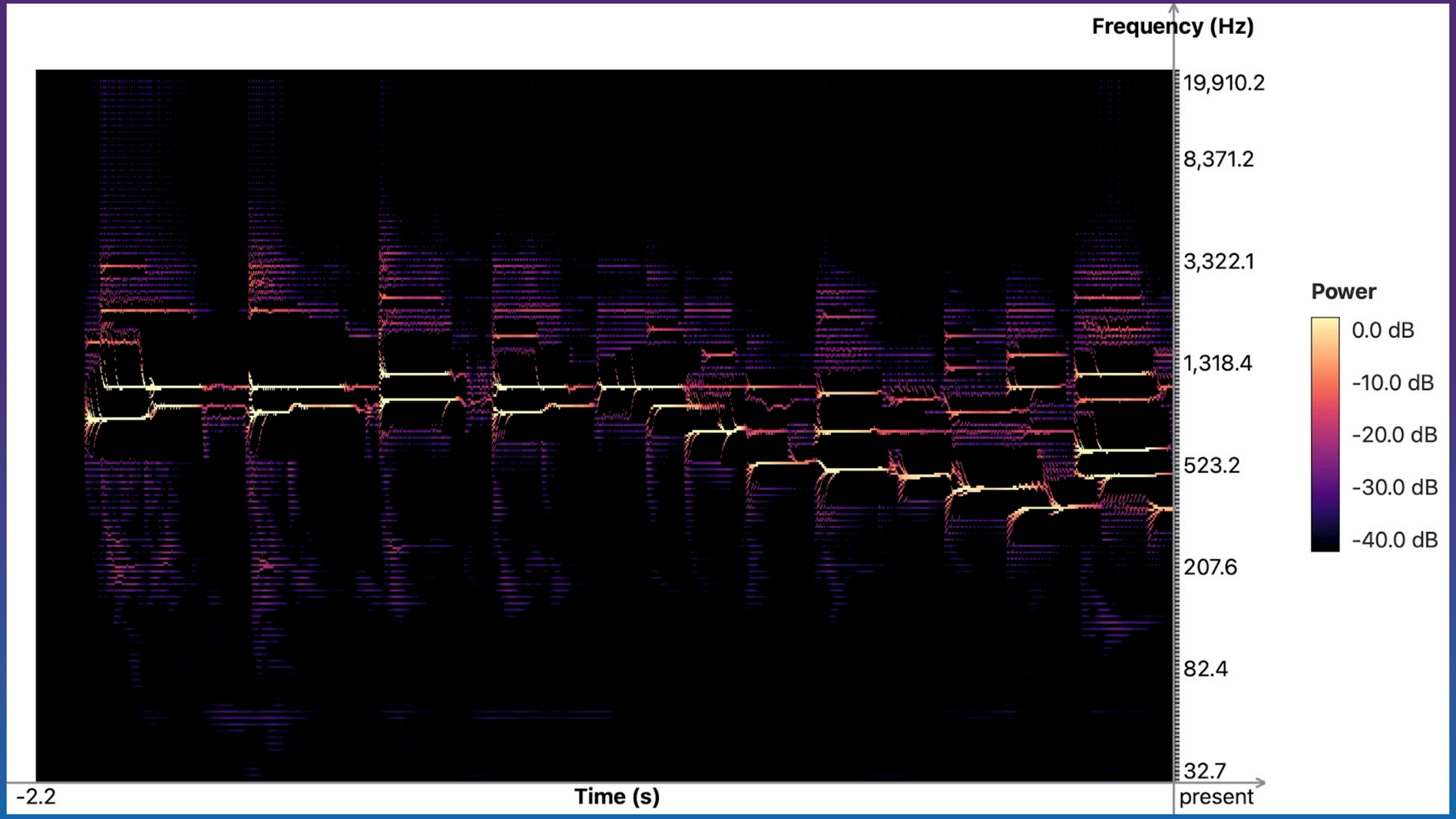
Outline

- Intro
- Resonate model
- Phase matters - instantaneous frequencies
- Self-tuning banks
- **Frequency tracking spectrograms**
- Synthesis
- Outro

© 2026 Alexandre R. J. François

Tracking Spectrogram

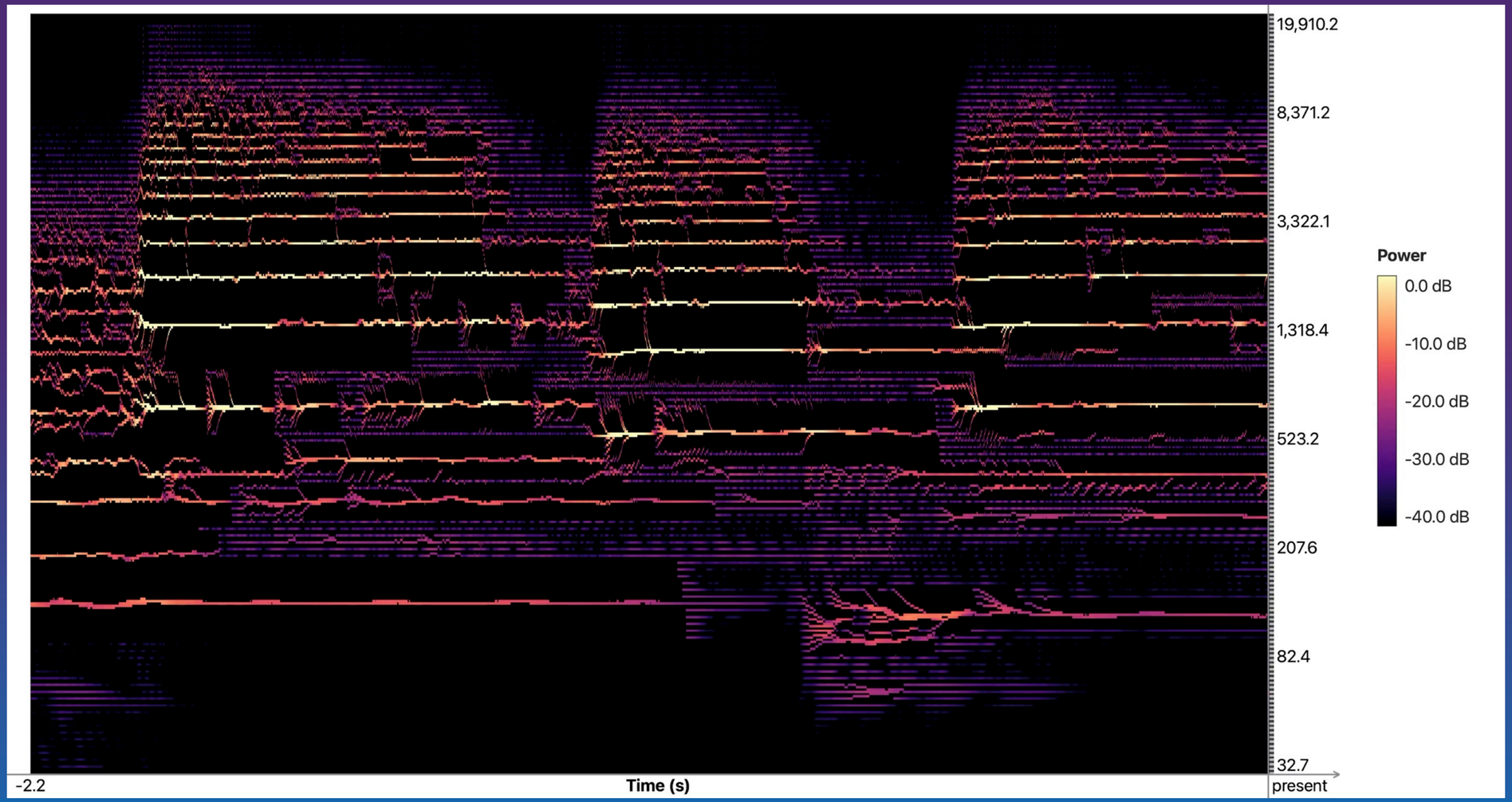
Who’s Loving You - The Jacksons 5



© 2026 Alexandre R. J. François

Tracking Spectrogram

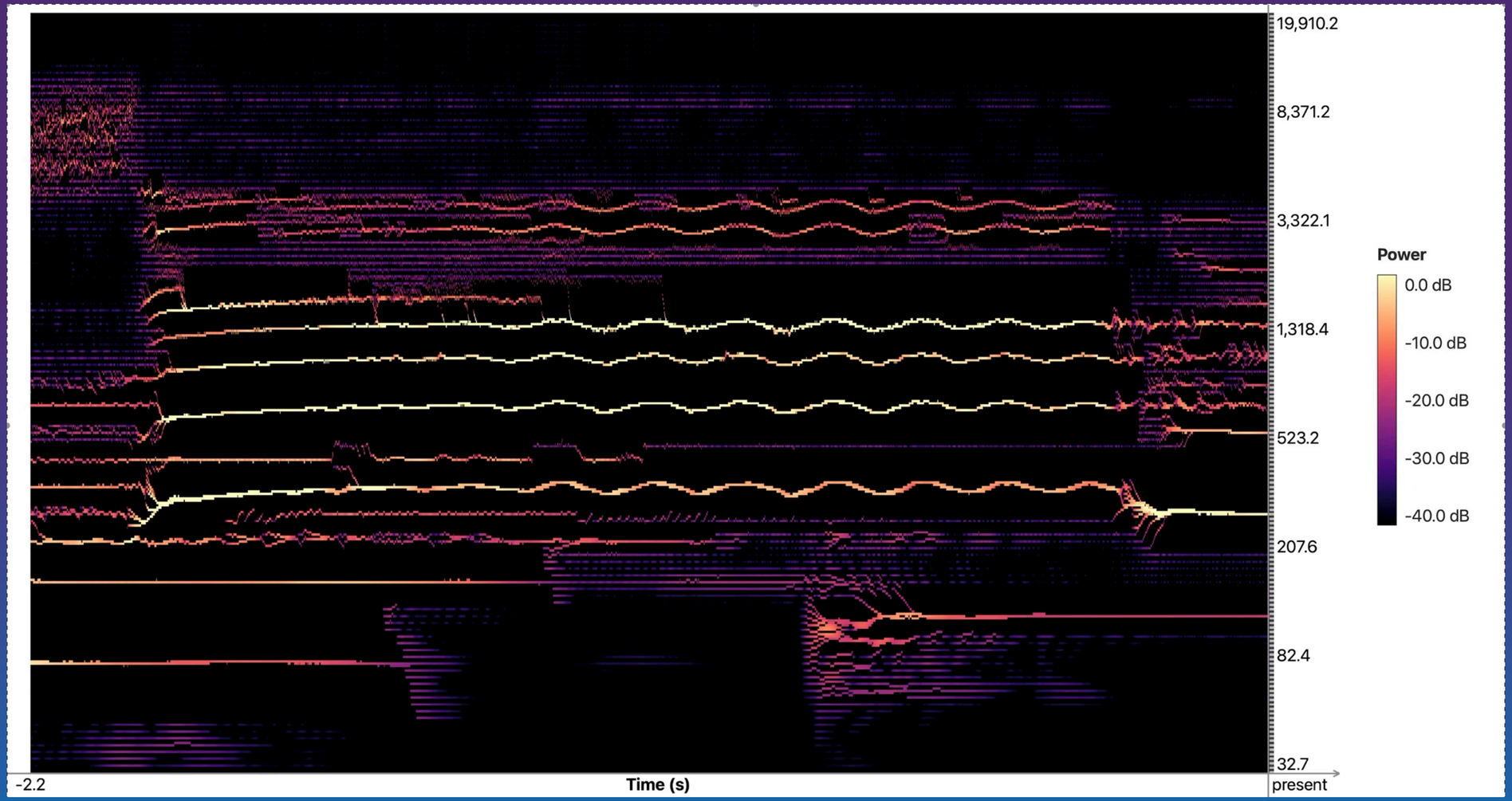
Summertime - Louis Armstrong



© 2026 Alexandre R. J. François

Tracking Spectrogram

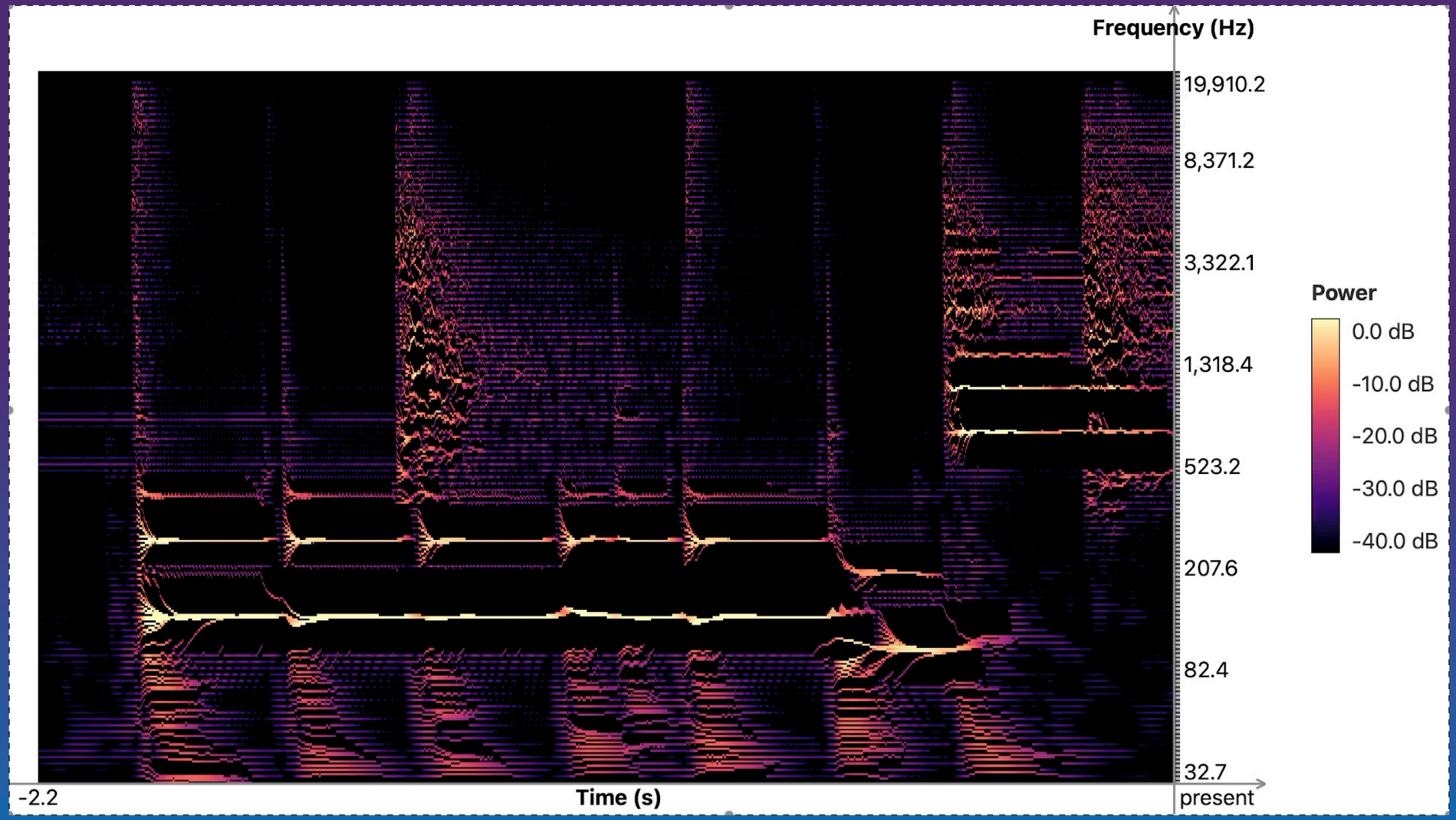
Summertime - Ella Fitzgerald



© 2026 Alexandre R. J. François

Tracking Spectrogram

Under Pressure - Queen + Bowie



© 2026 Alexandre R. J. François

Outline

- Intro
- Resonate model
- Phase matters - instantaneous frequencies
- Self-tuning banks
- Frequency tracking spectrograms
- **Synthesis**
- Applications

© 2026 Alexandre R. J. François

Synthesis

- Inverse phasor
  - Frequency ratio: ρ
  - Sampling rate $f'_s = 1/\Delta t'$ (time manipulation)
- Additive synthesis
  - Sum non-overlapping tracking resonators: vertical phase coherence not an issue (no need for phase locking)
  - Quality challenges: transient smearing, partial reconstruction, other possible phase discontinuities, etc.

$$P_k^{-1}(t) = P_k^{-1}(t - \Delta t')e^{i\omega_k(t)\rho\Delta t'}$$

$$S_k(t) = \tilde{R}_k(t)P_k^{-1}(t)$$

$$s(t) = KRe(\sum_{k \in T(t)} S_k(t))$$

© 2026 Alexandre R. J. François

Synthesis Examples

Who’s Loving You

- **No time/frequency manipulation**
- Semi-tone up
- Semi-tone down
- Slower
- Faster

© 2026 Alexandre R. J. François

Synthesis Examples

Who’s Loving You

- No time/frequency manipulation
- **Semi-tone up**
- Semi-tone down
- Slower
- Faster

© 2026 Alexandre R. J. François

Synthesis Examples

Who’s Loving You

- No time/frequency manipulation
- Semi-tone up
- **Semi-tone down**
- Slower
- Faster

© 2026 Alexandre R. J. François

Synthesis Examples

Who’s Loving You

- No time/frequency manipulation
- Semi-tone up
- Semi-tone down
- **Slower**
- Faster

© 2026 Alexandre R. J. François

Synthesis Examples

Who’s Loving You

- No time/frequency manipulation
- Semi-tone up
- Semi-tone down
- Slower
- **Faster**

© 2026 Alexandre R. J. François

Outline

- Intro
- Resonate model
- Phase matters - instantaneous frequencies
- Self-tuning banks
- Frequency tracking spectrograms
- Synthesis
- **Outro**

© 2026 Alexandre R. J. François

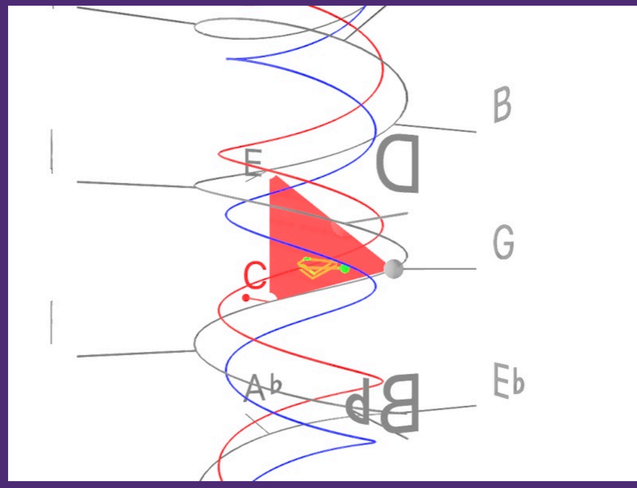
Summary

Known concepts applied in a different context

- Resonate model: efficient implementations, suitable for real-time, low latency applications
- Based on known techniques, avoids many of the complexities inherent to FFT/iFFT, but poses different questions
  - Dynamics: EWMA parameters
  - Non dominant frequency components?
  - Bank composition (application dependent)
    - Number of resonators?
    - Natural frequencies distribution, medium term plasticity?

© 2026 Alexandre R. J. François

Applications
What's next?



- Output is much more powerful than spectrograms capture
- Better audio features?
  - Tracking Chromagram -> MuSA_RT
  - Timbre analysis, better than MFCCs?
  - Onset detection?
  - Audio -> MIDI? Beat tracking?
- Non verbal audio control e.g. for accessibility, games, etc.?
- **Causal systems cannot know the future: model and feed back expectation, at different time scales**
- ML: stop trying to "learn" what we already know?

© 2026 Alexandre R. J. François

Thank you!



www.alexandrefrancois.org/Resonate

© 2026 Alexandre R. J. François