Sparse Random Wavelet Signal Representation and Decomposition

Nicholas Richardson

BMath Math Physics '20

MMath Applied Math '22

June 1st, 2021

UNIVERSITY OF WATERLOO

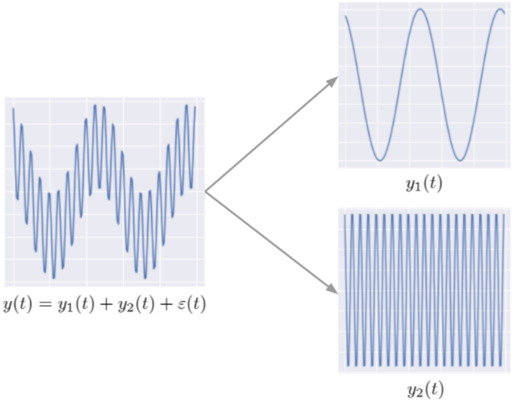
Joint work with Giang Tran, Hayden Schaeffer & Rachel Ward

Outline

Outline

Problem Statement

Given an input signal y, decompose it into simple modes.



Want meaningful modes, ex. mode for each acoustic source or type of oscillation

And why...?

For signal processing:

- Analysis on simple modes may be easier than the full signal
- Label/identification of modes
- Separate noise from signal (denoising)
- Pre-processing step

For music decomposition specifically:

- Make a karaoke or a cappella track
- Study or re-mix songs (live recordings, historical)
- For easier melodic/harmonic analysis, instrument labelling, or transcription

Existing Methods

And their downsides

- **Fourier Series/Transform**: time information not captured
- **Empirical Mode Decomposition**: only decomposes into intrinsic mode functions
- **Short-Time Fourier Transform**: dense representation, input must be evenly sampled
- **Neural Networks**: slow, requires large training set

Outline

Our Method

"Sparse Random Wavelet Signal Representation and Decomposition"

- *Signal*: 1D Function $y(t) : [0, t_{\max}] \to \mathbb{R}$
- *Wavelet*: $g_m(t) = \text{window}(t; m) \cdot e^{if_m t}$
- *Random*: wavelets are picked randomly
- *Representation*: $y(t) \approx \sum_m x_m g_m(t)$
- *Sparse*: most of x_m are zero
- *Decomposition*: write $y(t)$ as a sum of modes $\sum_k y_k(t)$

Method Overview

1. Find sparse representation for y
   - Compress data
   - Create clear distinctions between modes
2. Cluster nearby representation elements
   - Assume modes are connected regions

Advantages:

- Fast & efficient: only input signal is required
- Blind: no side information of signal required
- Can handle unevenly sampled data

Outline

Input

Input: $\{t_i, y_i\}_{i=1}^{N}$

Given N samples $\{y_i\}$ of a function $y(t)$ at times $\{t_i\}$

Note the t_i's do not have to be equally spaced.

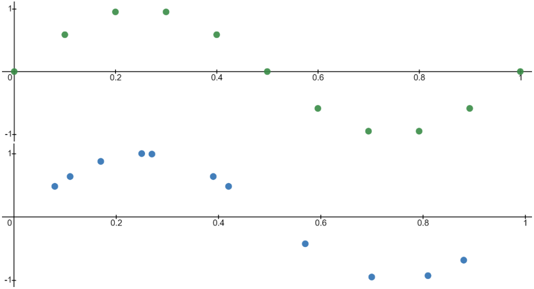


Figure: Equally Spaced (top) vs. Random Sampling (bottom)

Basis Functions

Real Gabor Wavelet

$$g_m(t) = e^{-2\frac{(t-\tau_m)^2}{w^2}} \sin(2\pi f_m \cdot t + \phi_m),$$
$$\tau_m \in \mathcal{U}(0, t_{\max}), f_m \in \mathcal{U}(0, f_{\max}), \phi_m \in \mathcal{U}(0, 2\pi) \tag{1}$$

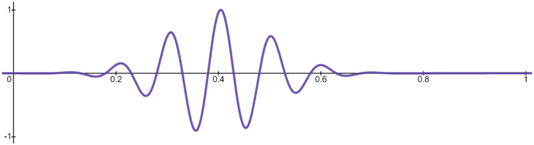


Figure: Example wavelet $(\tau, f, \phi) = (.2, 10, 1.26)$

Representation Algorithm

via Basis Pursuit Denoising

Algorithm: Sparse Wavelet Representation for y

input: $t, y \in \mathbb{R}^N, M \in \mathbb{Z}^+, w, f_{\text{max}} \in \mathbb{R}^+, r \in]0,1[$

Generate $\{\tau_m, f_m, \phi_m\}_{m=1}^{M}$;

Generate M wavelets $g_m \in \mathbb{R}^N$;

Store wavelets by column in a matrix $G \in \mathbb{R}^{N \times M}$;

Solve $x^* = \underset{x \in \mathbb{R}^M}{\arg\min} \|x\|_1$ s.t. $\|Gx - y\|_2 < \sigma = r\|y\|_2$;

output: $x^*, G, \{\tau_m, f_m, \phi_m\}_{m=1}^{M}$

- L_1 norm promotes sparsity
- Optimization is solved via Python's SPGL1 package

Reconstruction

Reconstruct y by multiplying wavelets with weights found

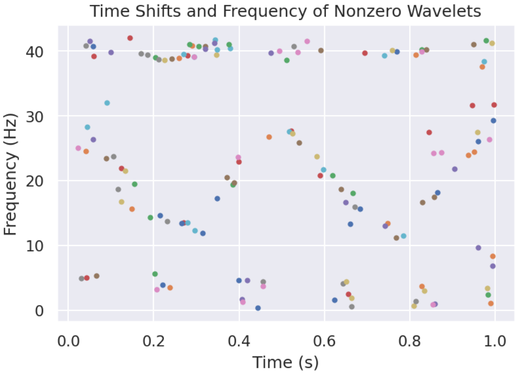
$$y \approx Gx^* = \sum_{m=1}^{M} x_m g_m. \tag{2}$$

Relative error in this representation $\frac{\|Gx^* - y\|_2}{\|y\|_2}$ is less than r

Outline

Clustering

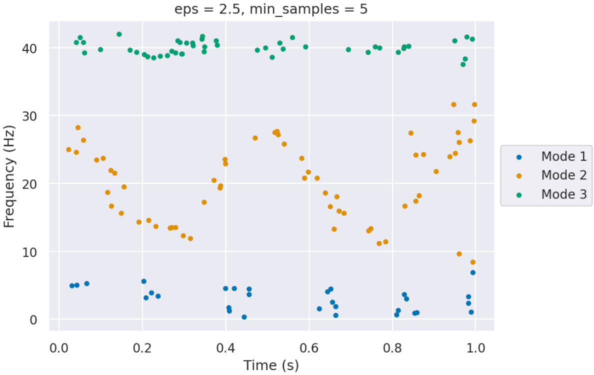
Plot $\{\tau_m, f_m\}_{m=1}^{M}$ where $x_m^* \neq 0$



Goal: cluster points that are close together

Clustering

Plot $\{\tau_m, f_m\}_{m=1}^{M}$ where $x_m^* \neq 0$



Goal: cluster points that are close together

Decomposition Algorithm

Algorithm: Sparse Representation Decomposition

input: $x^*, G, (\tau_m, f_m)_{m=1}^M$, $\texttt{min\_samples} \in \mathbb{Z}^+$, $\epsilon, s \in \mathbb{R}^+$

Define $(\tau_{m_j}, f_{m_j})_{j=1}^{M'} := \{(\tau_m, f_m) | x_m^* \neq 0\}$;

Scale input points to obtain $(\tau_{m_j}, s \cdot f_{m_j})$;

Use DBSCAN to label each point by cluster to obtain $\{\ell_{m_j}\}$
where $\ell_{m_j} \in \{-1, 0, \cdots, K-1\}$;

Extract K modes: $y_k = \sum_{m \in I_k} x_m^* g_m,\ I_k = \{m_j | \ell_{m_j} = k-1\}$;

output: $\{y_k\}_{k=1}^{K'}$

- Uses DBSCAN algorithm from scikit-learn's cluster module

Outline

Outline

Three Mode Decomposition 1

Uniquely Banded Modes, Evenly Sampled

Input: $y(t) = y_1(t) + y_2(t) + y_3(t)$ defined by,

$$\begin{aligned}
y_1(t) &= -2t + \sin(10\pi t + \phi_1) \\
y_2(t) &= \sin\left(2\pi\left(20t + \frac{2}{3}\sin(4\pi t)\right) + \phi_2\right) \\
y_3(t) &= (\sin(8\pi t) + 2)\sin\left(80\pi t + \phi_3\right),
\end{aligned} \tag{3}$$

where $\phi_i \in \mathcal{U}(0, 2\pi), t \in [0, 1]$, with $N = 160$ samples.

Parameters: $f_{\max} = 46\,\text{Hz}, M = 4416, w = 0.1\,\text{s}, r = 6\%$

$\texttt{min\_samples} = 5, \epsilon = 2.5, s = 1$

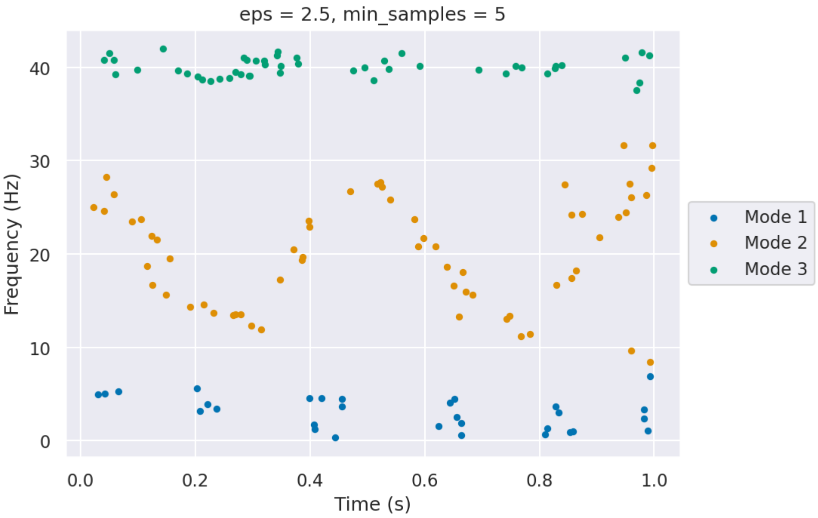


Figure: Segmentation of nonzero wavelets into three modes

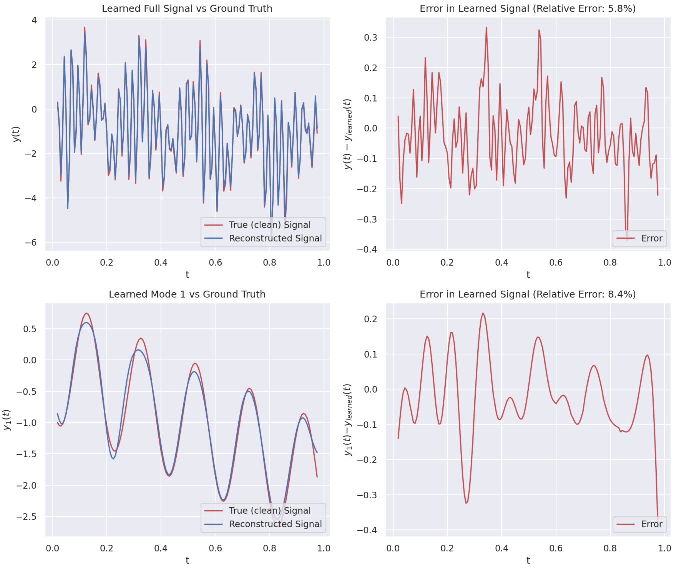


Figure: Decomposition results: full signal and mode 1 of 3 shown.

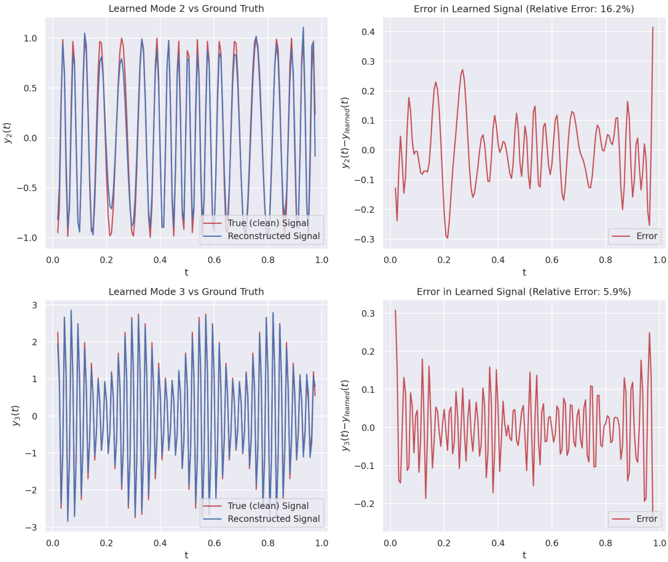


Figure: Decomposition results: modes 2 and 3 shown.

Three Mode Decomposition 2

Noncontinuous and overlapping, Evenly Sampled

Input: $y(t) = y_1(t) + y_2(t) + y_3(t)$ defined by

$$y_1(t) = \pi t,\ t \in [0, 5/4[$$

$$y_2(t) = \cos(40\pi t),\ t \in [0, 5/4[$$

$$y_3(t) = \cos\left(\frac{4}{3}\left((2\pi t - 10)^3 - (2\pi - 10)^3 + 20\pi(t-1)\right)\right),\ t \in]1, 2].$$

Parameters: $f_{\mathrm{max}} = 80, M = 10\,240, r = 5\%, w = 0.1\,\mathrm{s},$

$\mathtt{min\_samples} = 5, \epsilon = 0.125, s = 1/80.$

Synchrosqueezed wavelet transforms: An empirical mode decomposition-like tool; Daubechies, Lu, & Wu; Applied and Computational Harmonic Analysis, 2011. DOI:10.1016/j.acha.2010.08.002

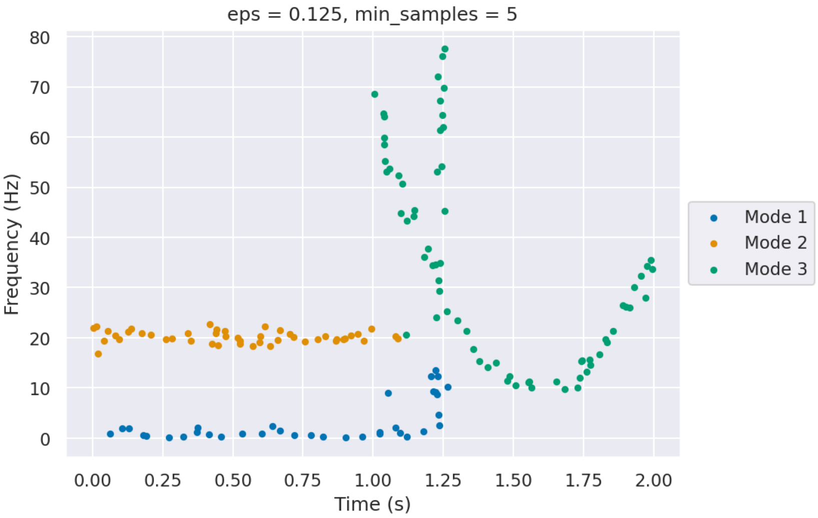


Figure: Segmentation of nonzero wavelets into three modes

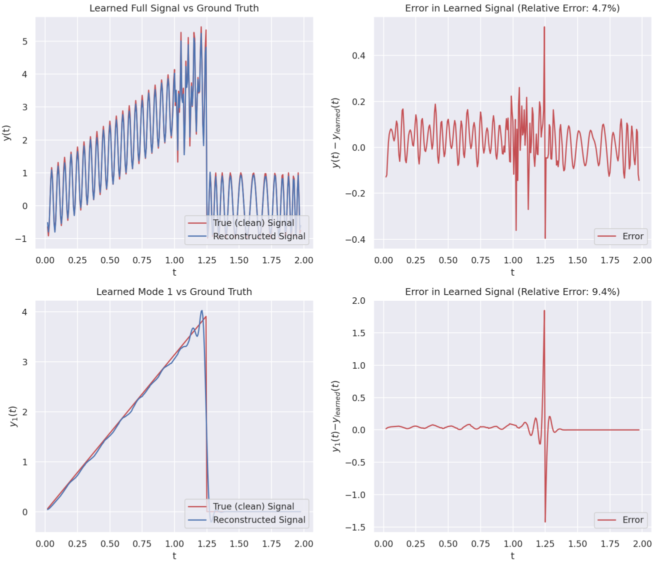


Figure: Decomposition results: full signal and mode 1 of 3 shown.

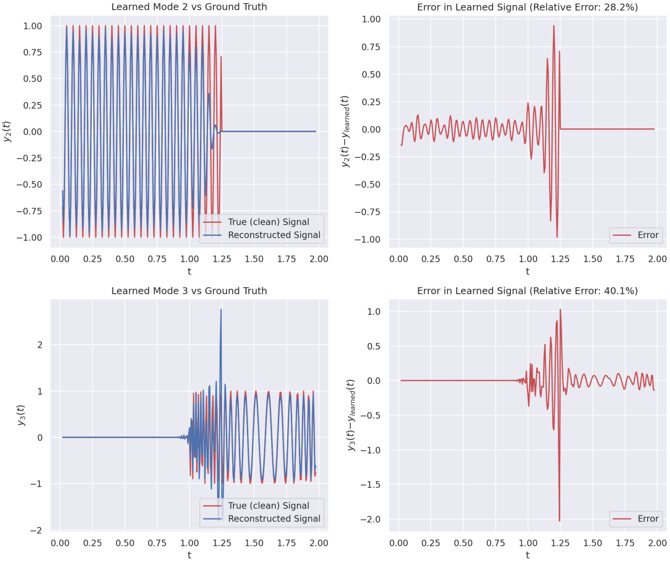


Figure: Decomposition results: modes 2 and 3 shown.

Outline

Flute and Guitar Decomposition

Input: $y(t) = y_1(t) + y_2(t)$ where $y_1(t)$ is a flute and $y_2(t)$ is a guitar clip around 1.85 s long.

$N = 44\,100\,\text{Hz}/8 \cdot 1.85\,\text{s} \approx 10\,200$ samples.

Parameters:

$f_{\text{max}} = 44\,100\,\text{Hz}/16$, $M = 100\,000$, $r = 8\%$, $w = 0.03\,\text{s}$,.

(play examples)

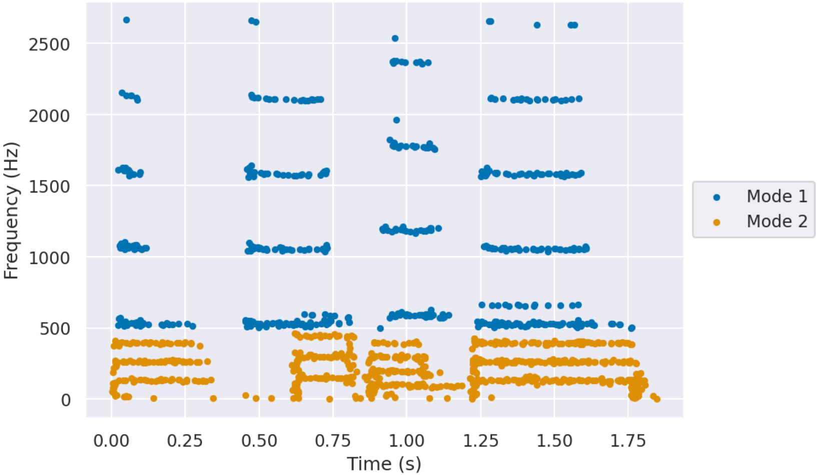


Figure: Segmentation via slice at $f = 480\,\mathrm{Hz}$

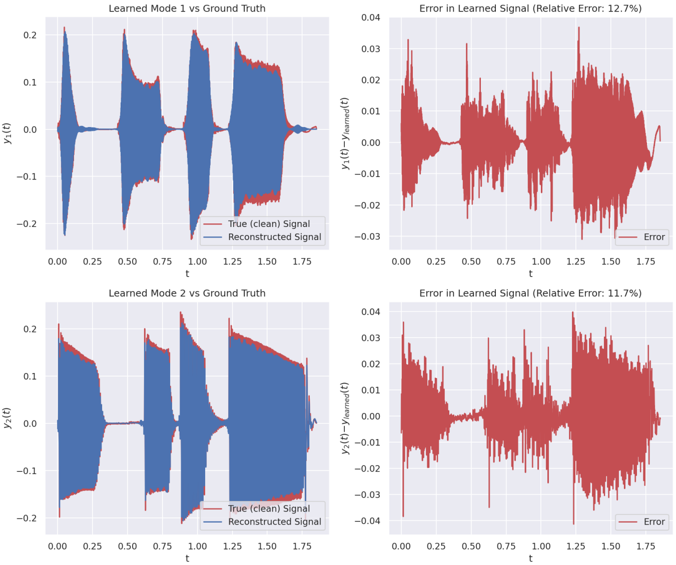


Figure: Decomposition results: modes 1 and 2 shown.

Outline

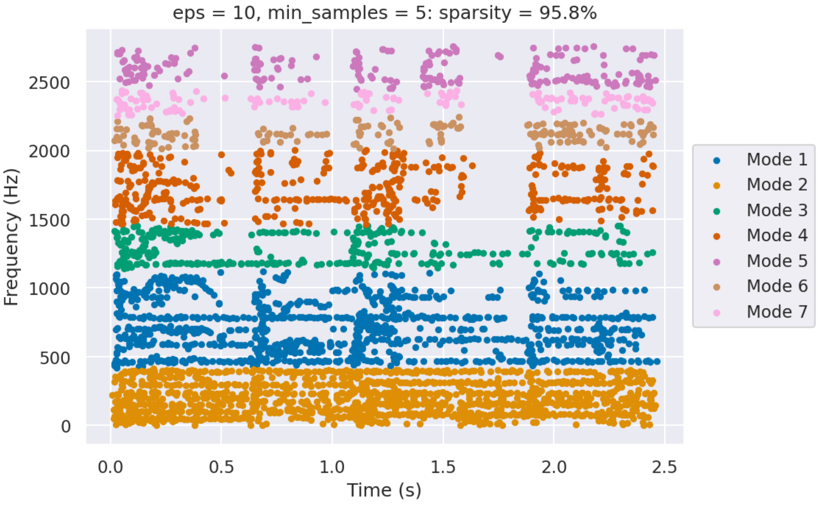


Figure: Attempted Full Song Decomposition

Suggested Improvements

Segmentation:

- Group wavelets with similar weights
- Use two step clustering to group harmonics/notes first
- Define new metric where integer multiple frequencies are considered close

Clustering:

- Higher level basis for more sparsity
- Experiment with LASSO to enforce a given level of sparsity

References

- Deep Learning for Audio Signal Processing

  Purwins et. al.

  arXiv:1905.00078

- Generalization Bounds for Sparse Random Feature Expansions

  Hashemi, Schaeffer, Shi, Topcu, Tran, & Ward

  arXiv:2103.03191

- Sparse Optimization with Least-Squares Constraints

  Louhichi et. al.

  DOI:10.1137/100785028

- A density based algorithm for discovering clusters with varied density

  van den Berg & Friedlander

  DOI:10.1109/WCCAIS.2014.6916622

Thank you for listening!

Nicholas Richardson

njericha@uwaterloo.ca

Code for Examples

Additional Decomposition Example

Uniquely Banded Modes, Randomly Sampled

Input: $y(t) = y_1(t) + y_2(t) + y_3(t)$ defined by,

$$
\begin{aligned}
y_1(t) &= -2t + \sin(10\pi t + \phi_1) \\
y_2(t) &= \sin\left(2\pi\left(20t + \frac{2}{3}\sin(4\pi t)\right) + \phi_2\right) \\
y_3(t) &= (\sin(8\pi t) + 2)\sin\left(80\pi t + \phi_3\right),
\end{aligned}
\tag{4}
$$

where $\phi_i \in \mathcal{U}(0, 2\pi)$, $t \in [0, 1]$, with $N = 160$ samples.

Parameters: $f_{\text{max}} = 46\,\text{Hz}$, $M = 4416$, $w = 0.1\,\text{s}$, $r = 6\%$

$\texttt{min\_samples} = 5, \epsilon = 2.5, s = 1$

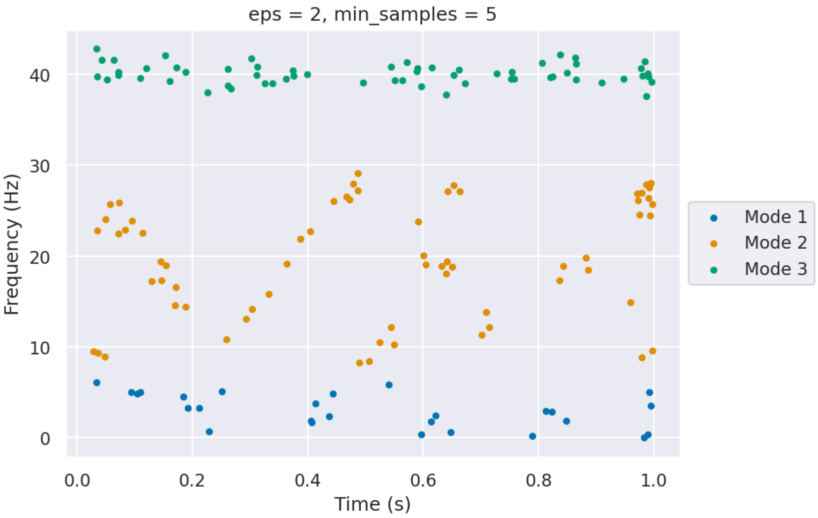


Figure: Segmentation of nonzero wavelets into three modes

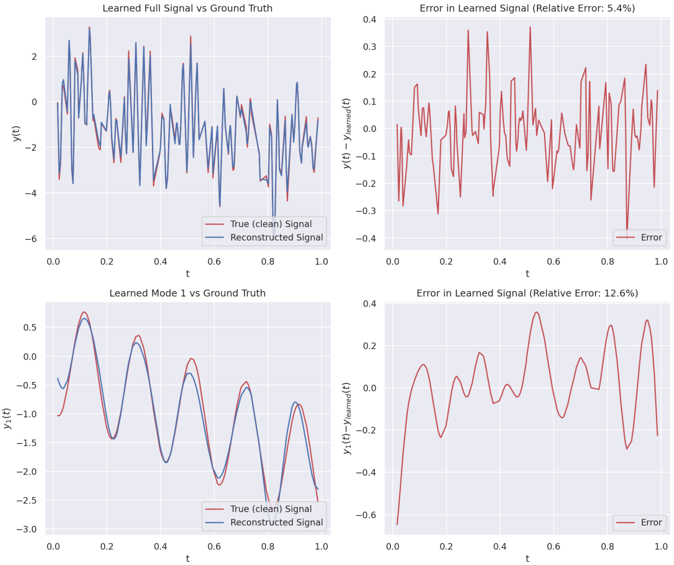


Figure: Decomposition results: full signal and mode 1 of 3 shown.

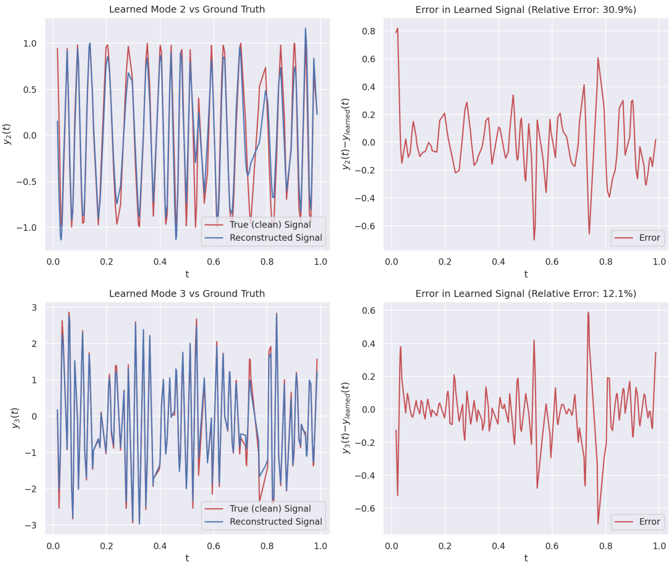


Figure: Decomposition results: modes 2 and 3 shown.

Flute and Guitar Decomposition

With Wavelet Clustering

Input: $y(t) = y_1(t) + y_2(t)$ where $y_1(t)$ is a flute and $y_2(t)$ is a guitar clip around 1.85 s long.

$N = 44\,100\,\text{Hz}/8 \cdot 1.85\,\text{s} \approx 10\,200$ samples.

Parameters:

$f_{\text{max}} = 44\,100\,\text{Hz}/16$, $M = 100\,000$, $r = 8\%$, $w = 0.03\,\text{s}$,

$\texttt{min\_samples} = 5, \epsilon = 350, s = 1.$

(play examples)

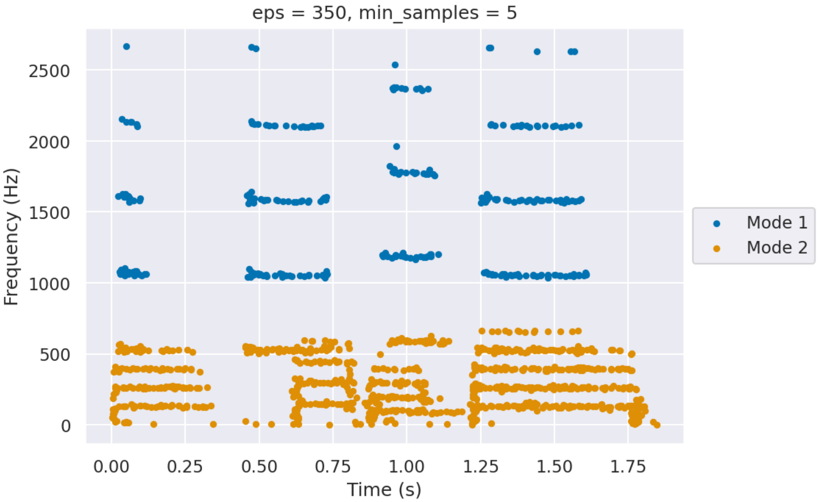


Figure: Segmentation of nonzero wavelets into two modes

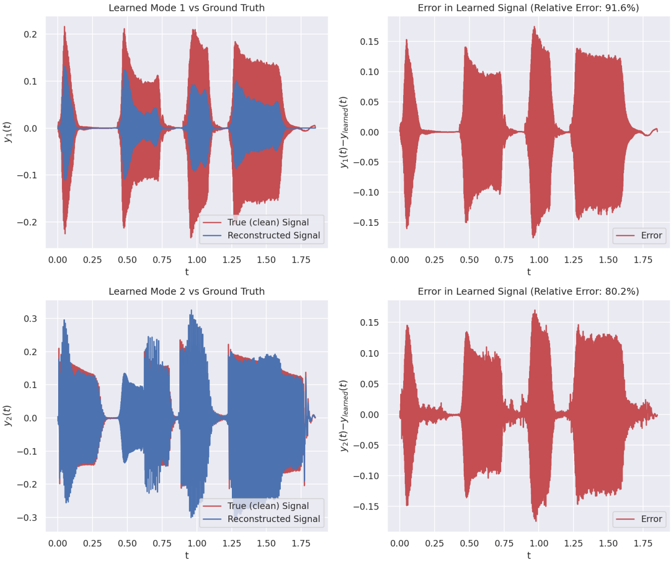


Figure: Decomposition results: modes 1 and 2 shown.