Iterative Filtering Algorithms

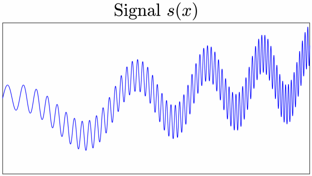
Giovanni Barbarino
Department of Mathematics
and Systems Analysis,
Aalto University

Antonio Cicone
Department of Information Engineering
Computer Science and Mathematics,
University of L'Aquila

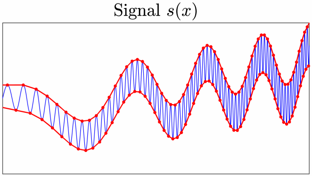
NoSAG21 Conference, L'Aquila

Model of Iterative Filtering Algorithms

Empirical Mode Decomposition

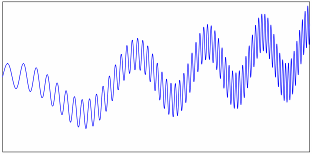


Empirical Mode Decomposition

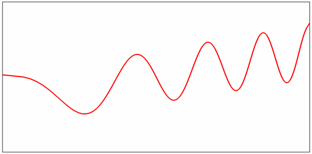


Empirical Mode Decomposition

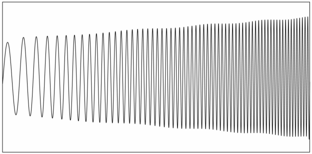
Signal $s(x)$



Moving Average $\mathcal{L}(s)$

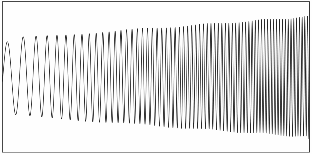


$\mathcal{S}(s) = s - \mathcal{L}(s)$

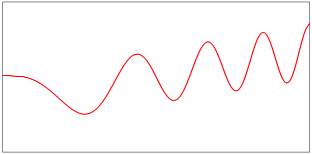


Empirical Mode Decomposition

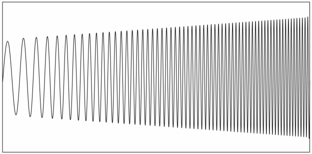
$\mathcal{S}(s)$



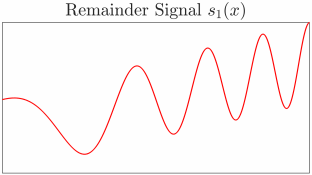
$\mathcal{L}(s) + \mathcal{L}(\mathcal{S}(s))$

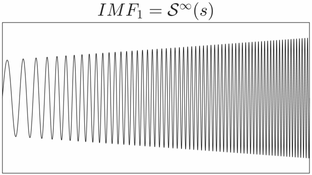


$\mathcal{S}^2(s)$



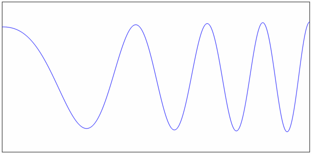
Empirical Mode Decomposition



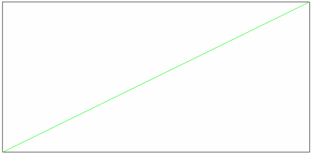


Empirical Mode Decomposition

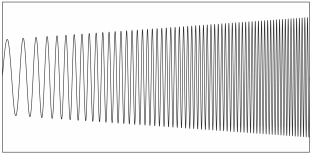
IMF_2



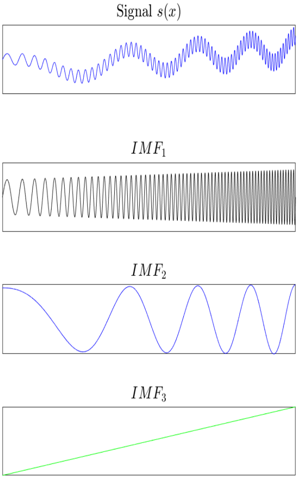
IMF_3



IMF_1



Empirical Mode Decomposition



Decomposition of non-stationary signals into Intrinsic Mode Functions (IMF)

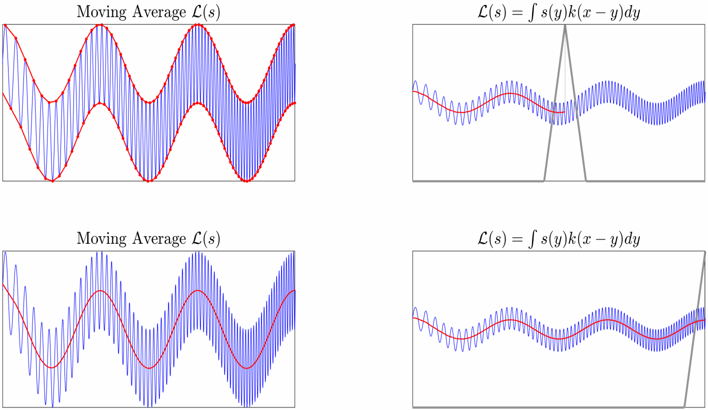
- Iterative Method
- Based on the computation of the moving average of the signal
- Splits the signal into simple oscillatory components

Numerous variants (EEMD, NA-MEMD, FMEMD, etc.) have been proposed in the years to deal with instability and mode splitting/mixing, and to prove its convergence

Empirical Mode Decomposition

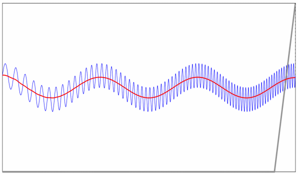
The effect of the moving average is to flatten the highest frequency component

A way to emulate the effect is to use a filter on the signal

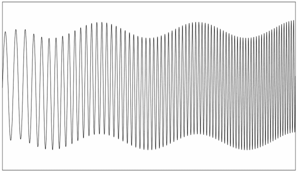


Iterative Filtering

$\mathcal{L}(s) = \int s(y)k(x-y)dy$



$\mathcal{S}(s) = s(x) - \int s(y)k(x-y)dy$



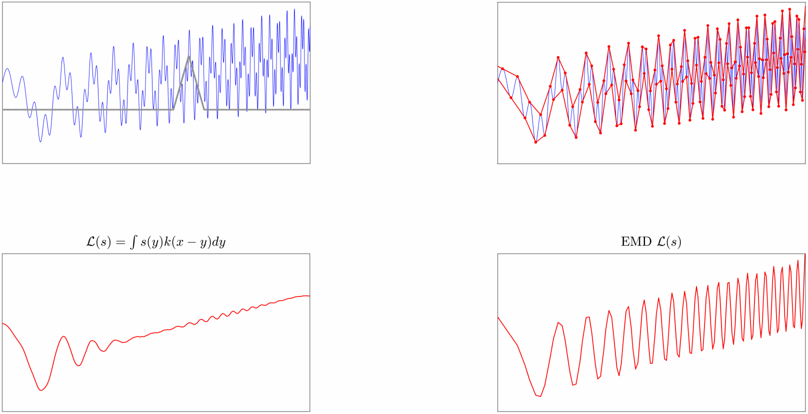
Choose the filter k:

- Unit-norm, even, nonnegative and compact supported
- $k = \omega \star \omega$
- Smooth

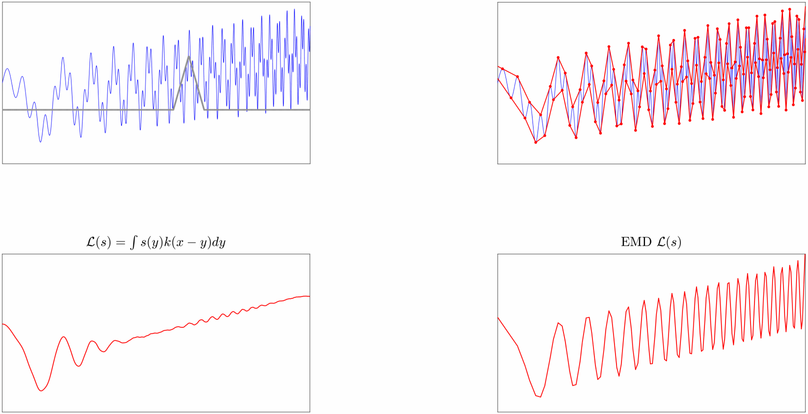
The IF method iteratively apply the filter through convolution

$$\mathcal{S}(f) := f(x) - \int f(y)k(x-y)dy$$
$$IMF = IMF \cup \{\mathcal{S}^{\infty}(s)\}$$
$$s = s - \mathcal{S}^{\infty}(s)$$

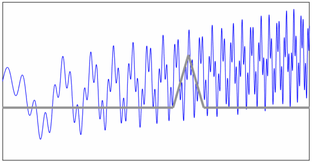
$\mathcal{S}^{\infty}(s)$ always converges and the method is fast (cyclic matrix, FFT), but it is not as flexible as EMD...



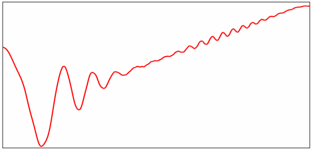
Let's take a look at the instantaneous frequencies



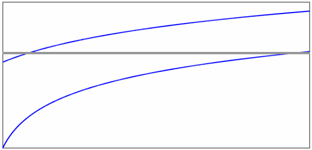
Let's take a look at the instantaneous frequencies



$\mathcal{L}(s) = \int s(y)k(x-y)dy$



Instantaneous Frequencies

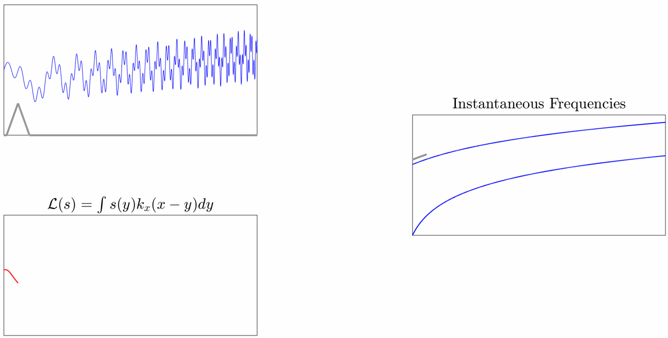


$$\hat{\mathcal{L}(s)} = \hat{s}(y) \cdot \hat{k}(y)$$

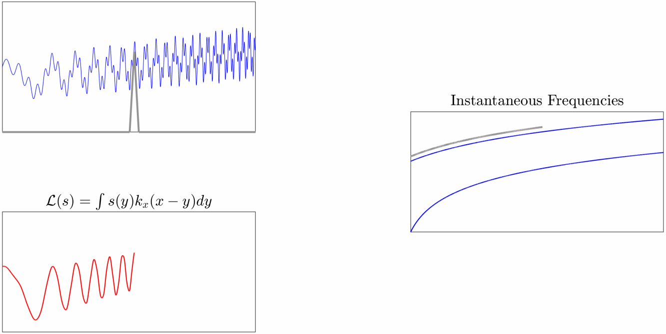
IF does not work with non-disjoint bands of frequencies

Adaptive Local Iterative Filtering

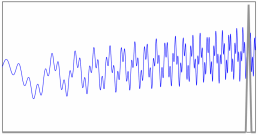
$$k_x(y) := k(\ell(x)^{-1}y)\ell(x)^{-1}$$



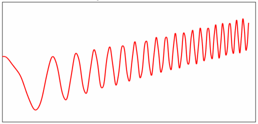
$$k_x(y) := k(\ell(x)^{-1}y)\ell(x)^{-1}$$



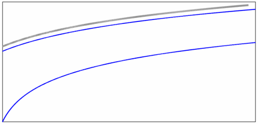
$$k_x(y) := k(\ell(x)^{-1}y)\ell(x)^{-1}$$



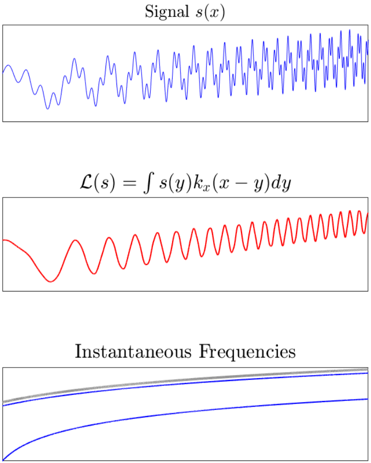
$\mathcal{L}(s) = \int s(y)k_x(x-y)dy$



Instantaneous Frequencies



Adaptive Local Iterative Filtering



Given the signal $s(x)$, fix the filter

$$k_x(y) := k(\ell(x)^{-1} y)\ell(x)^{-1}$$

and apply iteratively the filter through convolution

$$\mathcal{S}(f) := f(x) - \int f(y) k_x(x-y) dy$$
$$IMF = IMF \cup \{\mathcal{S}^\infty(s)\}$$
$$s = s - \mathcal{S}^\infty(s)$$

ALIF is now as flexible as EMD, and empirically converges, but..

- No structure, not fast as IF ($O(n^2)$ against $O(n)$)
- Has no clean formal analysis
- $\mathcal{S}^\infty(s)$ is not always convergent (in the discrete setting)

Discrete Setting

$$\boldsymbol{s} = [s(h)\ s(2h)\ \dots\ s(1-h)\ s(1)] \qquad h = 1/N$$

$$s(x) - \int_0^1 s(y) k_x(x-y) dy|_{x=ah} \quad \sim \quad \boldsymbol{s}_a - \frac{1}{N} \sum_{b=1}^{N} k\left(\frac{(a-b)h}{\ell(ah)}\right) \frac{1}{\ell(ah)} \boldsymbol{s}_b$$

$\mathcal{S}(\boldsymbol{s}) := \boldsymbol{s} - M\boldsymbol{s} = (I - M)\boldsymbol{s}$

- $\mathcal{S}^\infty(\boldsymbol{s})$ converges when

$$|\lambda_i(I - M)| < 1 \vee \lambda_i(I - M) = 1$$

- Converges to the kernel of M

The kernel is the same in αM where $\alpha \in \mathbb{R}$, so the real condition is

$$\Im(\lambda_i(M)) > 0 \vee \lambda_i(M) = 0$$

Setting a stopping condition in the iteration makes $\mathcal{S}^\infty(\boldsymbol{s})$ a near-kernel vector

For big enough N and if $\ell(x)$ is continuous, positive and

$$k(x) = \omega(x) \star \omega(x),$$

then the spectrum of M respects the condition for almost every eigenvalue

There are artificial examples where M has negative eigenvalues, so the convergence is not always assured

Discrete Setting

$$\boldsymbol{s} = [s(h)\ s(2h)\ \dots\ s(1-h)\ s(1)] \qquad h = 1/N$$

$$s(x) - \int_0^1 s(y) k_x(x-y) dy|_{x=ah} \quad \sim \quad \boldsymbol{s}_a - \frac{1}{N} \sum_{b=1}^{N} k\left(\frac{(a-b)h}{\ell(ah)}\right) \frac{1}{\ell(ah)} \boldsymbol{s}_b$$

$\mathcal{S}(\boldsymbol{s}) := \boldsymbol{s} - M\boldsymbol{s} = (I - M)\boldsymbol{s}$

- $\mathcal{S}^\infty(\boldsymbol{s})$ converges when

$$|\lambda_i(I-M)| < 1 \vee \lambda_i(I-M) = 1$$

- Converges to the kernel of M

The kernel is the same in αM where $\alpha \in \mathbb{R}$, so the real condition is

$$\Im(\lambda_i(M)) > 0 \vee \lambda_i(M) = 0$$

Setting a stopping condition in the iteration makes $\mathcal{S}^\infty(\boldsymbol{s})$ a near-kernel vector

For big enough N and if $\ell(x)$ is continuous, positive and

$$k(x) = \omega(x) \star \omega(x),$$

then the spectrum of M respects the condition for almost every eigenvalue

There are artificial examples where M has negative eigenvalues, so the convergence is not always assured

Discrete Setting

$$\mathbf{s} = [s(h)\ s(2h)\ \ldots\ s(1-h)\ s(1)] \qquad h = 1/N$$

$$s(x) - \int_0^1 s(y) k_x(x-y) dy|_{x=ah} \quad \sim \quad \mathbf{s}_a - \frac{1}{N} \sum_{b=1}^{N} k\left(\frac{(a-b)h}{\ell(ah)}\right) \frac{1}{\ell(ah)} \mathbf{s}_b$$

$\mathcal{S}(\mathbf{s}) := \mathbf{s} - M\mathbf{s} = (I - M)\mathbf{s}$

- $\mathcal{S}^\infty(\mathbf{s})$ converges when

$$|\lambda_i(I - M)| < 1 \vee \lambda_i(I - M) = 1$$

- Converges to the kernel of M

The kernel is the same in αM where $\alpha \in \mathbb{R}$, so the real condition is

$$\Im(\lambda_i(M)) > 0 \vee \lambda_i(M) = 0$$

Setting a stopping condition in the iteration makes $\mathcal{S}^\infty(\mathbf{s})$ a near-kernel vector

For big enough N and if $\ell(x)$ is continuous, positive and

$$k(x) = \omega(x) \star \omega(x),$$

then the spectrum of M respects the condition for almost every eigenvalue

There are artificial examples where M has negative eigenvalues, so the convergence is not always assured

Stable ALIF

Given the ALIF matrix M, let

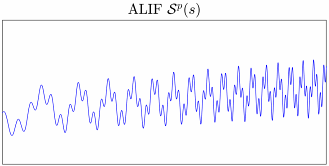
$$\mathcal{S}(\mathrm{s}) := \mathrm{s} - M^T M \mathrm{s} = (I - M^T M)\mathrm{s}$$

- $M^T M$ Has the same kernel of M
- $\lambda_i(M^T M) \geq 0$

As a consequence, $\mathcal{S}^\infty(\mathrm{s})$ always converges, but the method is way slower

- The cost per iteration is doubled
- There are more eigenvalues close to zero, so it takes more iterations to extract the exact component

$N = 3000$



Stable ALIF

Given the ALIF matrix M, let

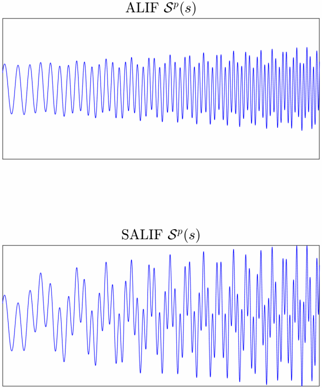
$$\mathcal{S}(\mathrm{s}) := \mathrm{s} - M^T M\mathrm{s} = (I - M^T M)\mathrm{s}$$

- $M^T M$ Has the same kernel of M
- $\lambda_i(M^T M) \geq 0$

As a consequence, $\mathcal{S}^\infty(\mathrm{s})$ always converges, but the method is way slower

- The cost per iteration is doubled
- There are more eigenvalues close to zero, so it takes more iterations to extract the exact component

$T = 1/20$ $N = 3000$



Stable ALIF

Given the ALIF matrix M, let

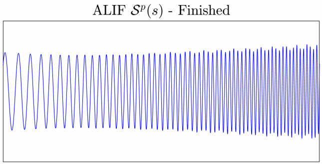
$$\mathcal{S}(\mathrm{s}) := \mathrm{s} - M^T M \mathrm{s} = (I - M^T M)\mathrm{s}$$

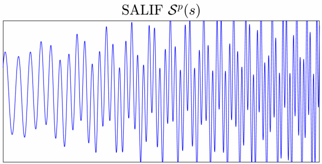
- $M^T M$ Has the same kernel of M
- $\lambda_i(M^T M) \geq 0$

As a consequence, $\mathcal{S}^\infty(\mathrm{s})$ always converges, but the method is way slower

- The cost per iteration is doubled
- There are more eigenvalues close to zero, so it takes more iterations to extract the exact component

$T = 2/20$ $N = 3000$





Stable ALIF

Given the ALIF matrix M, let

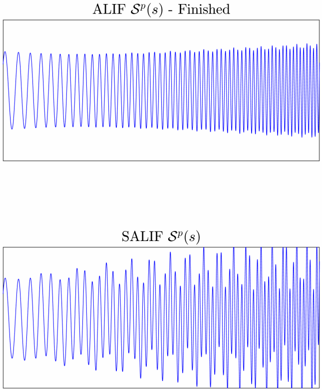
$$\mathcal{S}(\mathrm{s}) := \mathrm{s} - M^T M \mathrm{s} = (I - M^T M)\mathrm{s}$$

- $M^T M$ Has the same kernel of M
- $\lambda_i(M^T M) \geq 0$

As a consequence, $\mathcal{S}^\infty(\mathrm{s})$ always converges, but the method is way slower

- The cost per iteration is doubled
- There are more eigenvalues close to zero, so it takes more iterations to extract the exact component

$T = 3/20$ $N = 3000$



Stable ALIF

Given the ALIF matrix M, let

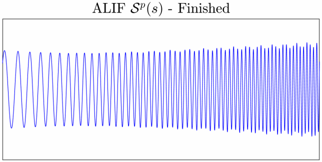
$$\mathcal{S}(\mathrm{s}) := \mathrm{s} - M^T M \mathrm{s} = (I - M^T M)\mathrm{s}$$

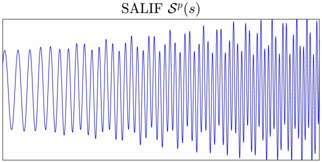
- $M^T M$ Has the same kernel of M
- $\lambda_i(M^T M) \geq 0$

As a consequence, $\mathcal{S}^\infty(\mathrm{s})$ always converges, but the method is way slower

- The cost per iteration is doubled
- There are more eigenvalues close to zero, so it takes more iterations to extract the exact component

$T = 4/20$ $N = 3000$





Stable ALIF

Given the ALIF matrix M, let

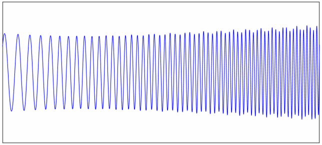
$$\mathcal{S}(\mathrm{s}) := \mathrm{s} - M^T M \mathrm{s} = (I - M^T M)\mathrm{s}$$

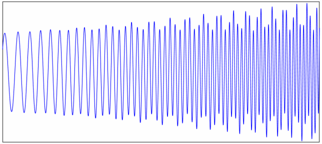
- $M^T M$ Has the same kernel of M
- $\lambda_i(M^T M) \geq 0$

As a consequence, $\mathcal{S}^\infty(\mathrm{s})$ always converges, but the method is way slower

- The cost per iteration is doubled
- There are more eigenvalues close to zero, so it takes more iterations to extract the exact component

$T = 5/20$ $N = 3000$

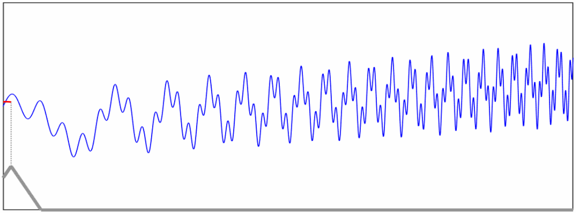




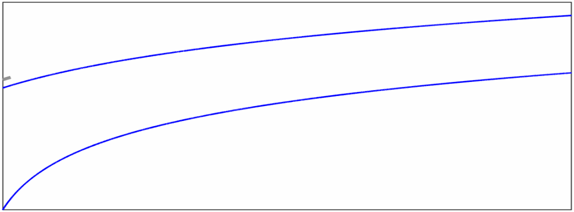
Resampled Iterative Filtering

ALIF

$$\mathcal{L}(s) = \int s(y)k_x(x-y)dy$$

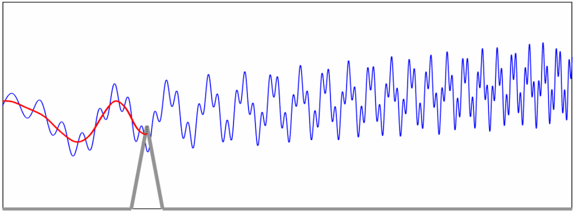


Instantaneous Frequencies

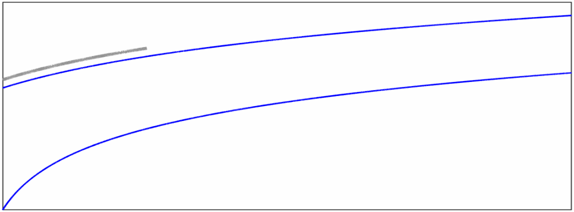


ALIF

$$\mathcal{L}(s) = \int s(y)k_x(x-y)dy$$

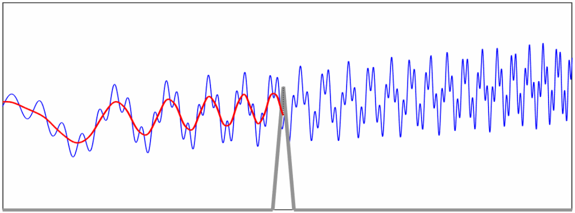


Instantaneous Frequencies

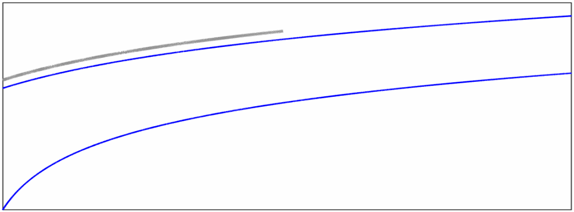


ALIF

$$\mathcal{L}(s) = \int s(y) k_x(x-y)dy$$

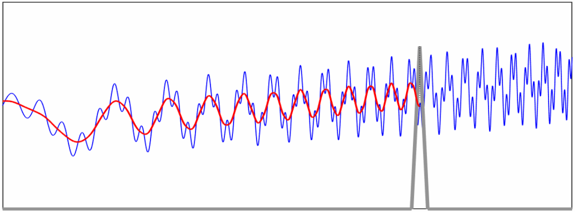


Instantaneous Frequencies

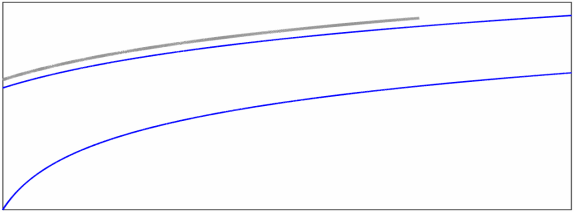


ALIF

$$\mathcal{L}(s) = \int s(y) k_x(x - y) dy$$

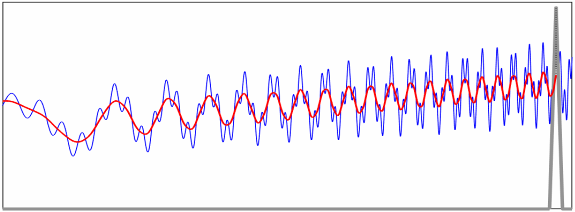


Instantaneous Frequencies

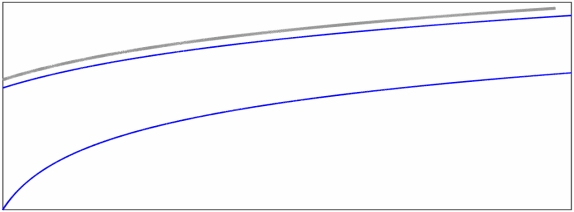


ALIF

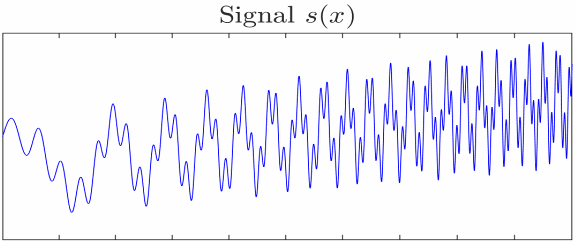
$$\mathcal{L}(s) = \int s(y) k_x(x-y) dy$$

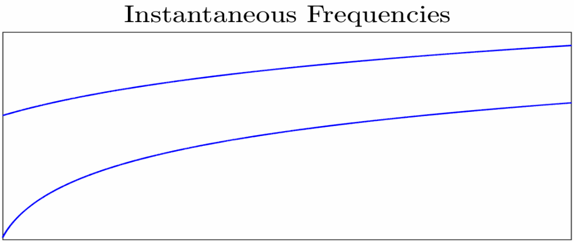


Instantaneous Frequencies

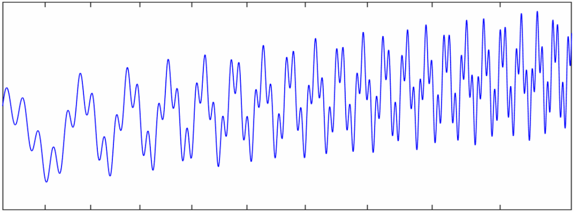


Resampling

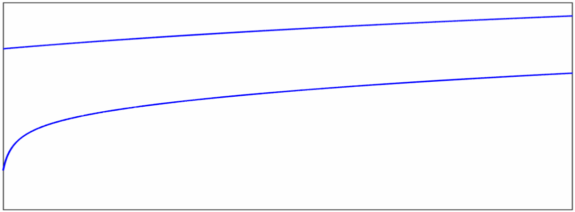




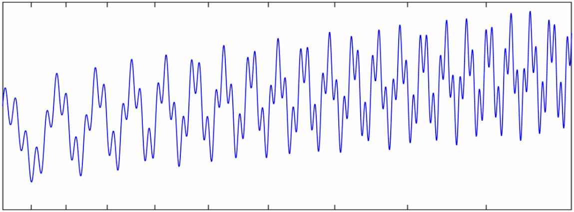
Resampling

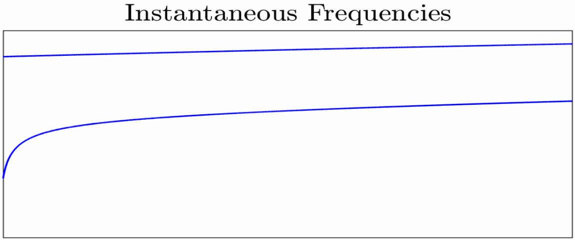


Instantaneous Frequencies

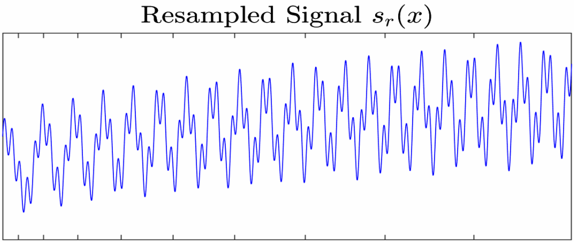


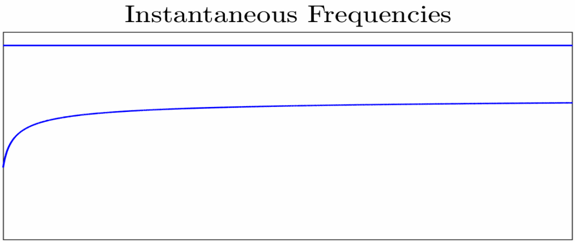
Resampling



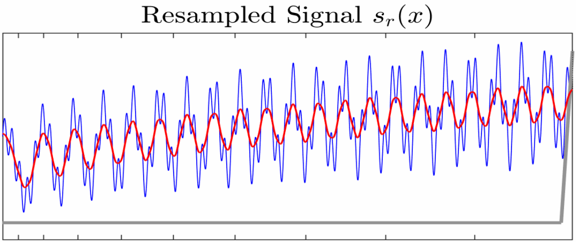


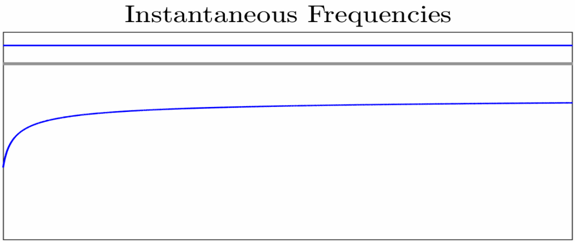
Resampling



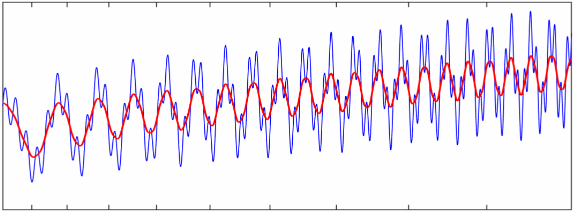


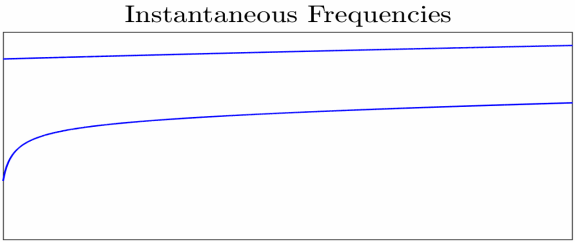
Resampling



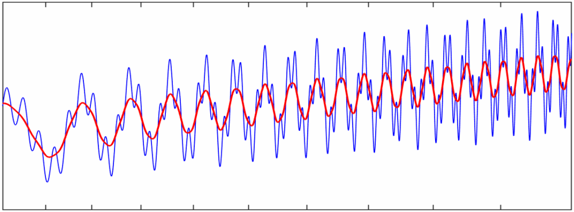


Resampling

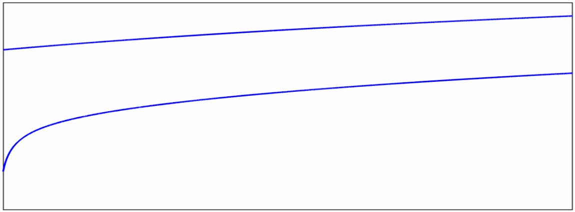




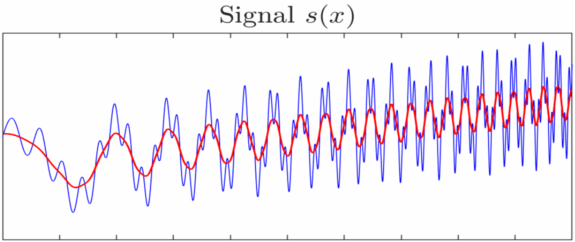
Resampling

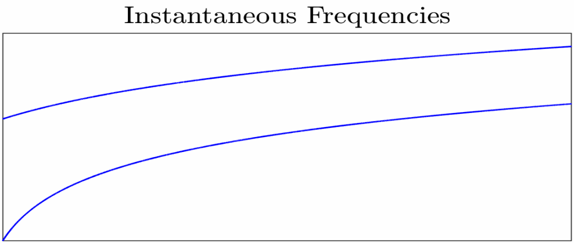


Instantaneous Frequencies



Resampling





Resampling Function $G(x)$

ALIF: $$\mathcal{S}(s)(x) := s(x) - \int s(y) k\left(\frac{x-y}{\ell(x)}\right)\frac{1}{\ell(x)}dy$$

$$t = (x-y)/\ell(x) \qquad x = G(z)$$

$$\mathcal{S}(s)(G(z)) := s(G(z)) - \int s(G(z) - t\ell(G(z)))k(t)\,dt$$

$$G'(z) = \ell(G(z)) \qquad G(z-t) \sim G(z) - tG'(z)$$

RIF: $$\mathcal{S}(s)(G(y)) := s(G(y)) - \int s(G(z-t))k(z)\,dz$$

ALIF is a "first order" RIF where

$$G^{-1}(z) = \int_0^z \frac{1}{\ell(x)}dx$$

ALIF $\neq$ RIF except when $\ell(x) = \ell$ and both are IF

Resampling Function $G(x)$

ALIF: $$\mathcal{S}(s)(x) := s(x) - \int s(y)k\left(\frac{x-y}{\ell(x)}\right)\frac{1}{\ell(x)}dy$$

$$t = (x-y)/\ell(x) \qquad x = G(z)$$

$$\mathcal{S}(s)(G(z)) := s(G(z)) - \int s(G(z) - t\ell(G(z)))k\,(t)\,dt$$

$$G'(z) = \ell(G(z)) \qquad G(z-t) \sim G(z) - tG'(z)$$

RIF: $$\mathcal{S}(s)(G(y)) := s(G(y)) - \int s(G(z-t))k\,(z)\,dz$$

ALIF is a "first order" RIF where

$$G^{-1}(z) = \int_0^z \frac{1}{\ell(x)}dx$$

ALIF $\neq$ RIF except when $\ell(x) = \ell$ and both are IF

Resampling Function $G(x)$

ALIF: $\mathcal{S}(s)(x) := s(x) - \int s(y) k\left(\frac{x-y}{\ell(x)}\right) \frac{1}{\ell(x)} dy$

$$t = (x-y)/\ell(x) \qquad x = G(z)$$

$$\mathcal{S}(s)(G(z)) := s(G(z)) - \int s(G(z) - t\ell(G(z))) k(t)\, dt$$

$$G'(z) = \ell(G(z)) \qquad G(z-t) \sim G(z) - tG'(z)$$

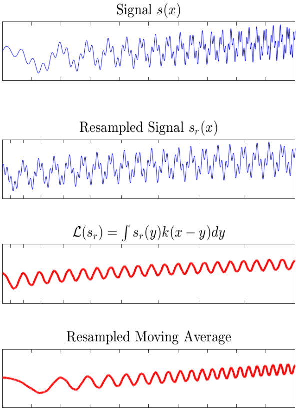
RIF: $\mathcal{S}(s)(G(y)) := s(G(y)) - \int s(G(z-t)) k(z)\, dz$

ALIF is a "first order" RIF where

$$G^{-1}(z) = \int_0^z \frac{1}{\ell(x)} dx$$

ALIF $\neq$ RIF except when $\ell(x) = \ell$ and both are IF

Resampled Iterative Filtering



Given the signal $s(x)$, compute the resampling

$$s_r(x) := s(G(x)) \qquad G^{-1}(z) = \int_0^z \frac{1}{\ell(x)}dx$$

and apply iteratively the filter through convolution

$$\mathcal{S}(f) := f(x) - \int f(y)k(x-y)dy$$

$$IMF = IMF \cup \{\mathcal{S}^\infty(s_r)(G^{-1}(x))\}$$

$$s = s - \mathcal{S}^\infty(s_r)(G^{-1}(x))$$

At the cost of two interpolations per IMF, we have an algorithm that is

- As flexible as ALIF
- Fast as IF, the resampling is outside the iterations
- $\mathcal{S}^\infty(s_r)$ is always convergent

Numerical Experiments

Experiment 1

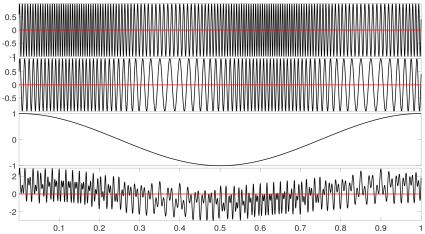
$N = 8000$

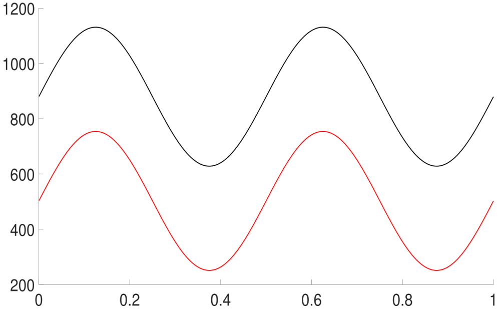
$h_1(x) = \cos(20\cos(4\pi t) - 160\pi t)$

$h_2(x) = \cos(20\cos(4\pi t) - 280\pi t)$

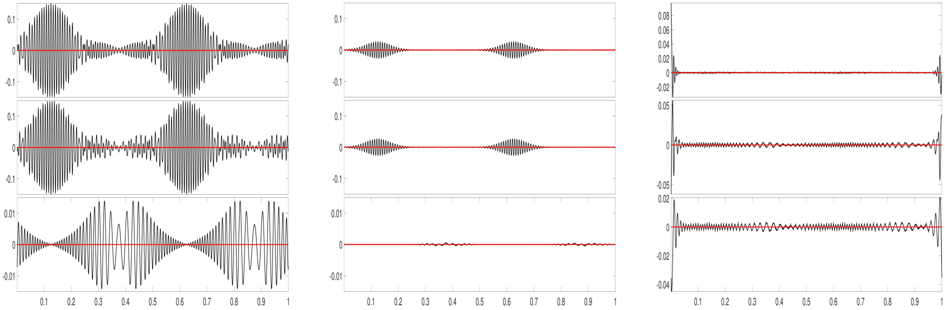
$h_3(x) = \cos(2\pi t)$

$h(x) = h_1(x) + h_2(x) + h_3(x)$



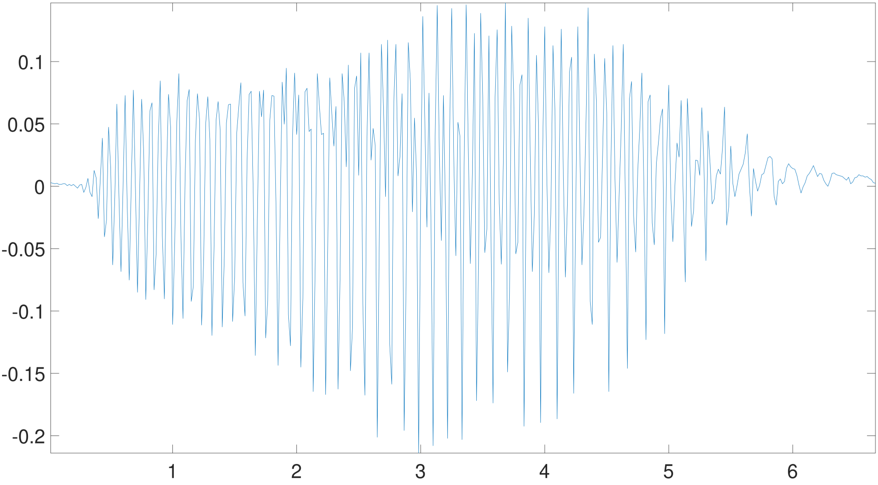


ALIF SALIF RIF

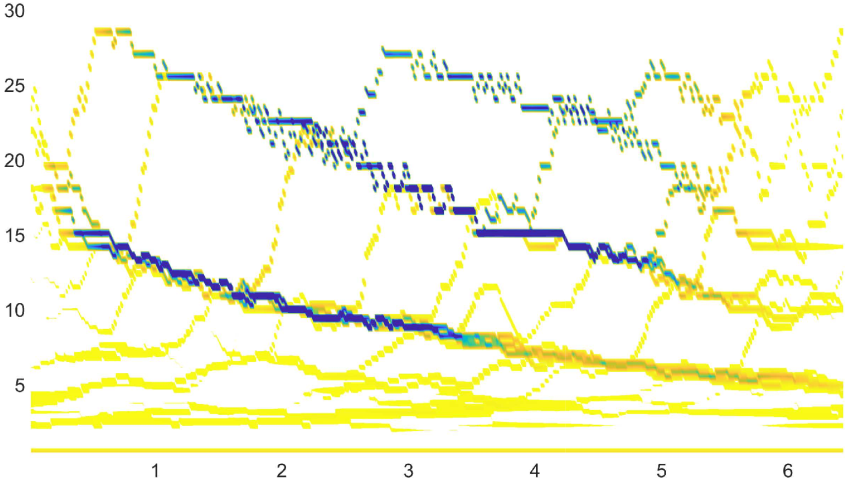


	Time	err1	err2	err3	Niter1	Niter2
ALIF	4.0860	0.070388	0.071158	0.008549	18	2
SALIF	19.7919	0.010054	0.010055	0.000161	353	5
RIF	1.4724	0.003426	0.003292	0.000908	81	11

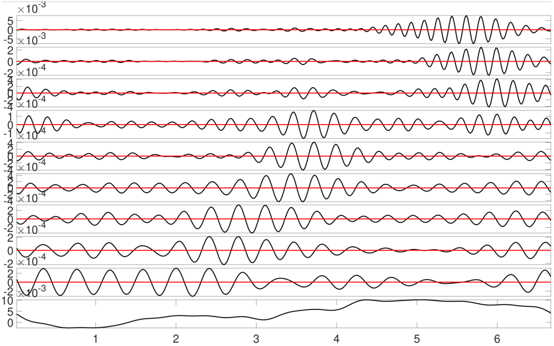
Experiment 2



Experiment 2

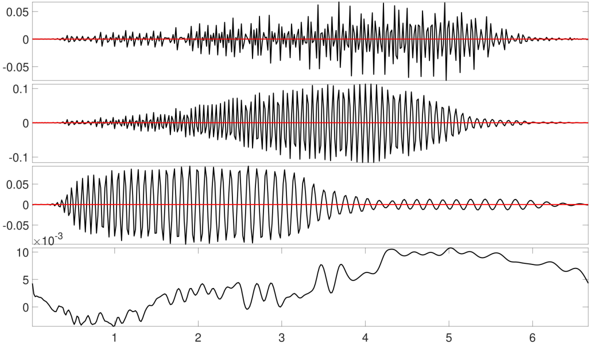


IF



×10^-3
5
0
-5
×10^-3
2
0
-2
×10^-4
×10^-4
1
0
-1
×10^-4
×10^-4
×10^-4
2
0
-2
×10^-4
2
0
-2
×10^-3
10
5
0
1
2
3
4
5
6

RIF



0.05
0
-0.05
0.1
0
-0.1
0.05
0
-0.05
×10^-3
10
5
0
1
2
3
4
5
6

Future Work

- More involved analysis (borders, length regularity, etc.)
- Multidimensional and Multi-Signals methods
- Direct computation of re-sampling
- Comparison with Synchroqueezing
- RIF/ALIF as denoising methods

Thank You!

- J. Liu A. Cicone and H. Zhou. **Adaptive local iterative filtering for signal decomposition and instantaneous frequency analysis.** *Applied and Computational Harmonic Analysis, 41(2)*, 2016.
- S. Serra-Capizzano A. Cicone, C. Garoni. **Spectral and convergence analysis of the discrete alif method.** *Linear Algebra and its Applications, 580*, 2019.
- A. Cicone G. Barbarino. **Conjectures on spectral properties of alif algorithm, https://arxiv.org/abs/2009.00582.** 2020.
- A. Cicone G. Barbarino. **Stabilization and variations to the alif algorithm: the fast resampled iterative filtering method.** (In preparation).
- Y. Wang L. Lin and H. Zhou. **Iterative filtering as an alternative algorithm for empirical mode decomposition.** *Advances in Adaptive Data Analysis, 1(4)*, 2009.
- S. R. Long M. C. Wu H. H. Shih Q. Zheng N.-C. Yen C. C. Tung N. E. Huang, Z. Shen and H. H. Liu. **The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis.** *Proceedings of the Royal Society of London. Series A: mathematical, physical and engineering sciences, 454(1971)*, 1998.