The Large Time-Frequency Analysis Toolbox

Zdeněk Průša

Austrian Academy of Sciences

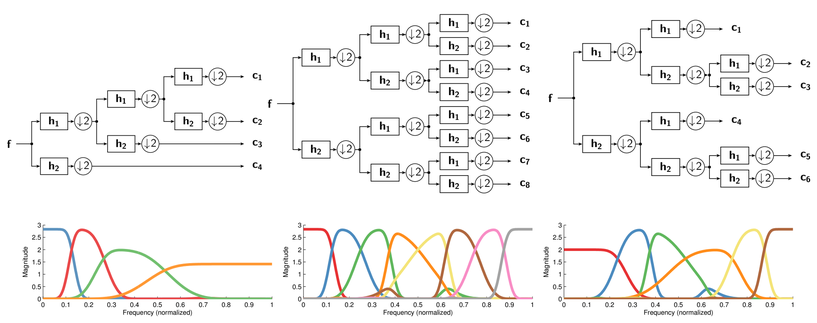
LTFAT is a Matlab/Octave toolbox for working with time-frequency **analysis**, **synthesis**, coefficient **manipulation** and **visualization**. It provides a large number of linear transforms including Gabor and wavelet transforms along with routines for constructing windows (filter prototypes). The toolbox is available freely under terms of GPLv3 at `http://ltfat.sourceforge.net`.

Toolbox overview

- Started by Peter L. Søndergaard in 2004.
- Current version 1.4.5.
- Official GNU Octave package since 1.4.2.
- Collection of TF transforms (and more):
  - Discrete Gabor transform
  - Windowed Modified Discrete Cosine transform, Disctete Wilson transform
  - Discrete Gabor transform with non-separable TF lattices
  - Nonstationary Discrete Gabor transform
  - Discrete wavelet transform and wavelet packets
  - General (oversampled) filterbanks
  - Constant-Q transform, Erblet transform
  - Spectrogram reassignment, window design
  - Classical Fourier analysis, signal processing, etc.
- MEX/OCT interfaces to a backend library in C language.
- **You are welcome to contribute!**

Discrete wavelet transform and wavelet packets

Fast wavelet transform and tools for creating and working with general filterbank trees (wavelet packets).



Best basis selection, M–band wavelets, wavelet frames, library of wavelet filters. Undecimated versions of the transforms. Non-tree-shaped identical filterbanks, routines for calculating frame bounds, canonical dual and tight frames of redundant systems.

Frames framework

- Object oriented abstraction layer unifying access to different transforms in LTFAT.
- Matrix algebra of frames in $\mathbb{C}^L$ **without matrices**.
- Fast algorithms whenever possible.

Columns of a $L \times \Lambda$ matrix $\mathbf{F}$ form a frame in $\mathbb{C}^L$ when frame bounds $0 < A \leq B < \infty$ exist with

$$A \left\|\mathbf{f}\right\|^2 \leq \left\|\mathbf{F}^*\mathbf{f}\right\|^2 \leq B \left\|\mathbf{f}\right\|^2$$

for all $f \in \mathbb{C}^L$. Optimum frame bounds: $A = \sigma^2_{\min}(\mathbf{F})$, $B = \sigma^2_{\max}(\mathbf{F})$ (squares of min. and max. singular values of $\mathbf{F}$).

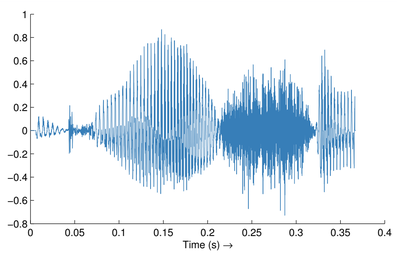
$\mathbf{F}$ has therefore full row rank, coordinates $\mathbf{c}$ of signal $\mathbf{f}$ in $\mathbf{F}$ such that $\mathbf{f} = \mathbf{F}\mathbf{c}$ are obtainable by right inversion of $\mathbf{F}$ as $\mathbf{c} = \mathbf{F}^*(\mathbf{F}\mathbf{F}^*)^{-1}\mathbf{f}$.

In LTFAT (no matrix)	Using matrices
`F = frame('type',params)`	$\mathbf{F}$
`c = frana(F,f)`	$\mathbf{c} = \mathbf{F}^*\mathbf{f}$
`f = frsyn(F,c)`	$\mathbf{f} = \mathbf{F}\mathbf{c}$
`Fd = framedual(F)`	$\mathbf{F}_\mathrm{d} = (\mathbf{F}\mathbf{F}^*)^{-1}\mathbf{F}$
`Ft = frametight(F)`	$\mathbf{F}_\mathrm{t} = (\mathbf{F}\mathbf{F}^*)^{-1/2}\mathbf{F}$
`framered(F)`	Λ/L
`framebounds(F)`	B/A
`[A,B]=framebounds(F)`	A, B
`c = franaiter(F,f)`	$\mathbf{c} \approx \mathbf{F}^*(\mathbf{F}\mathbf{F}^*)^{-1}\mathbf{f}$
`f = frsyniter(F,c)`	$\mathbf{f} \approx (\mathbf{F}\mathbf{F}^*)^{-1}\mathbf{F}\mathbf{c}$

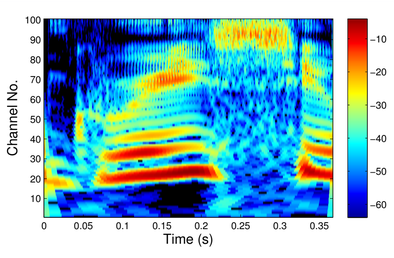
Remark:
`c = frana(framedual(F),f)` $\Rightarrow \mathbf{c} = \mathbf{F}^*(\mathbf{F}\mathbf{F}^*)^{-1}\mathbf{f}$
is known to be the solution of the problem:

$$\operatorname{argmin}_{\mathbf{c}} \ ||\mathbf{c}||_2, \qquad \text{subject to } \mathbf{F}\mathbf{c} = \mathbf{f}$$

Frames framework – test signal



A spoken word *greasy*. Recorded at 16 kHz, numer of samples is 5880.



Erblet spectrogram [3], frame type 'erbletfb'.

Frame analysis as the basis pursuit problem

Basis pursuit problem:

$$\operatorname{argmin}_{\mathbf{c}} \ ||\lambda\mathbf{c}||_1, \qquad \text{subject to } \mathbf{F}\mathbf{c} = \mathbf{f}$$

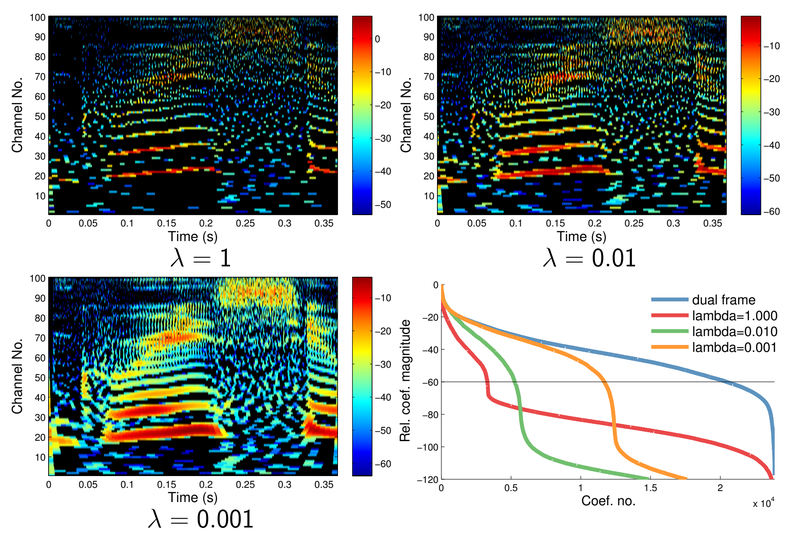
Split Augmented Lagrangian Shrinkage Algorithm – SALSA

- Initialize $\mathbf{c}, \mathbf{d}, \mu$
- repeat until stopping criterion is met
  - $\mathbf{v} \leftarrow \mathrm{soft}\left(\mathbf{c} + \mathbf{d}, \frac{\lambda}{\mu}\right) - \mathbf{d}$
  - $\mathbf{d} \leftarrow \mathbf{F}^*(\mathbf{F}\mathbf{F}^*)^{-1}(\mathbf{f} - \mathbf{F}\mathbf{v})$
  - $\mathbf{c} \leftarrow \mathbf{d} + \mathbf{v}$

Remark: $\mathbf{F}\mathbf{c} = \mathbf{f}$ holds in each iteration.

Basic usage:

```
c = franabp(F,f,lambda);
```



Frame analysis as the LASSO problem

LASSO or Basis pursuit denoising problem:

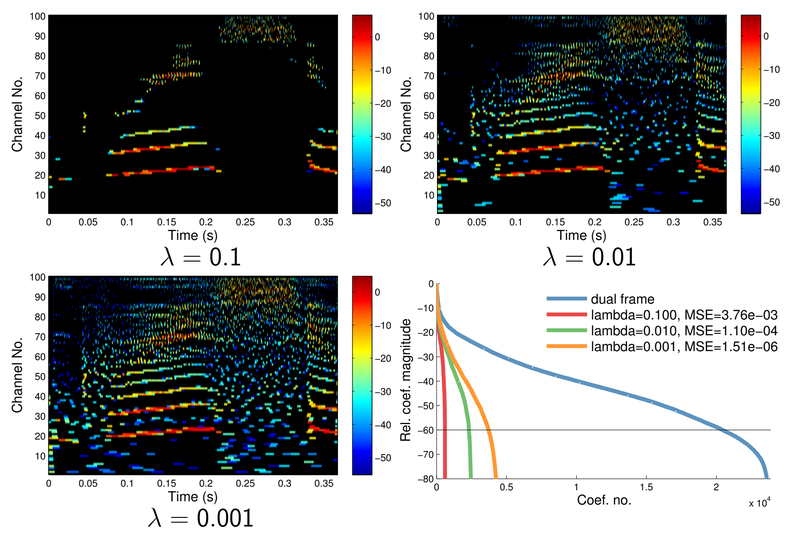
$$\operatorname{argmin}_{c} \ ||\lambda\mathbf{c}||_1 + \frac{1}{2}||\mathbf{F}\mathbf{c} - \mathbf{f}||_2^2$$

Fast Iterative Soft Thresholding Algorithm – FISTA

- Initialize $\mathbf{c}_0, \mathbf{z}, \mu, \tau_0 = 1, n = 1$
- repeat until stopping criterion is met
  - $\mathbf{c}_n \leftarrow \mathrm{soft}\left(\mathbf{z} + \frac{1}{\mu}\mathbf{F}^*(\mathbf{f} - \mathbf{F}\mathbf{z}), \frac{\lambda}{\mu}\right)$
  - $\tau_n \leftarrow \frac{1+\sqrt{1+4\tau_{n-1}^2}}{2}$
  - $\mathbf{z} \leftarrow \mathbf{c}_n + \frac{\tau_{n-1}-1}{\tau_n}(\mathbf{c}_n - \mathbf{c}_{n-1})$
  - $n \leftarrow n + 1$

Basic usage:

```
c = franalasso(F,f,lambda);
```



`frsynabs` – frame synthesis without a phase

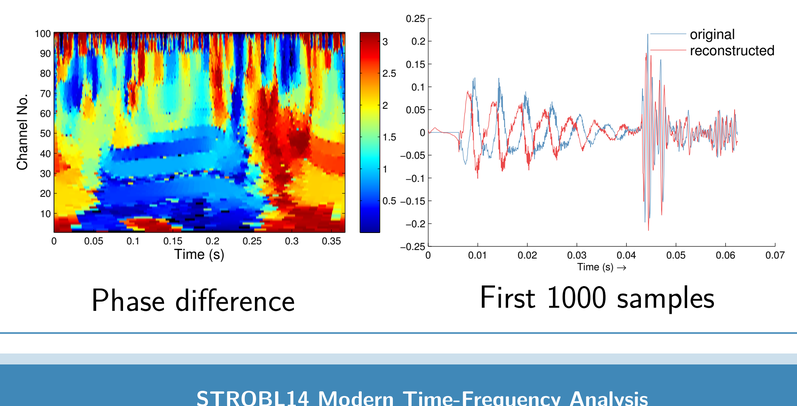
Problem: Find $\hat{\mathbf{f}}$ knowing $\mathbf{s} = |\mathbf{F}^*\mathbf{f}|$.

Griffin-Lim algorithm:

- Initialize $\boldsymbol{\varphi}$, $\mathbf{c} = \mathbf{s} \cdot \exp(i\boldsymbol{\varphi})$
- repeat until stopping criterion is met
  - $\boldsymbol{\varphi} \leftarrow \arg\left(\mathbf{F}^*(\mathbf{F}\mathbf{F}^*)^{-1}\mathbf{F}\mathbf{c}\right)$
  - $\mathbf{c} \leftarrow \mathbf{s} \cdot \exp\left(i\boldsymbol{\varphi}\right)$
- $\hat{\mathbf{f}} = (\mathbf{F}\mathbf{F}^*)^{-1}\mathbf{F}\mathbf{c}$

Basic usage:

```
f = frsynabs(F,s);
```



Phase difference

First 1000 samples

Blockstream processing programming interface

Allows the possibility for real-time audio processing directly in Matlab **and Octave**.
The block processing framework uses:

- **Portaudio** – a free, open-source, audio I/O C/C++ library. An uniform way of working with audio I/O across operating systems. `http://www.portaudio.com`
- **Playrec** – a MEX file providing an interface from Matlab to Portaudio. `http://github.com/PlayrecForMatlab`

Basic usage:

- `block(source)` – initializes the blockstream.
- `p=blockpanel(params)` – creates a control panel.

Repeat:

- `f=blockread()` – get next block of samples.
- `blockpanelget(p,params)` – get values from the panel.
- `blockplay(f)` – play (possibly modified) samples.

Blockstream sources

`block(source)` options:

- `'file.wav'` name of a wav file
- `'dialog'` shows a file-chooser dialog to choose a wav file.
- `'rec'` just record from a microphone/auxilary input.
- `'playrec'` record and play simultaneously.
- `data` input data as columns of a matrix for each input channel.

Some of the additional arguments:

- `'outfile','outfile.wav'` together with `blockwrite` performs on-the-fly writing to a file.
- `'offline'` initialization in offline mode.
- `'devid',[devIdOut,devIdIn]` specifies input and output audio devices.
- `'playch',[chanNo]` channels of the output device.
- `'recch',[chanNo]` channels of the input device.

Block analysis and synthesis

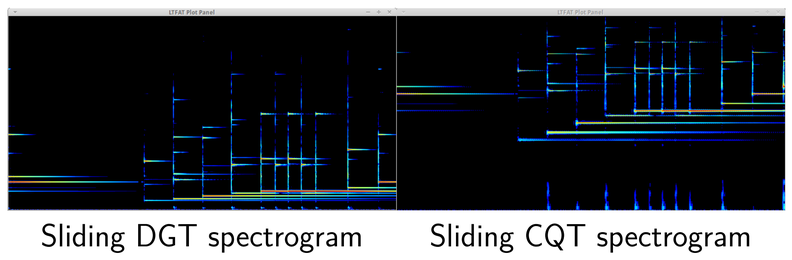
Main routines:

- `c=blockana(F,f)` – do frame analysis of a block of samples $\mathbf{f}$ using frame $\mathbf{F}$.
- `f=blocksyn(F,c)` – do frame synthesis of a block $\mathbf{f}$ using frame $\mathbf{F}$.

Three approaches to blockwise analysis and synthesis:

- **Direct** + works for most frames, + no additional delay − "bad" coefficients, − presence of block artifacts with coefficient modification.
- **Slicing window (metawindow)** + works for most frames, + reduces block artifacts, − coefficient modification is not straightforward.
- **Overlap-Save/Overlap-Add** + no blocking artifacts, + straightforward modification of coefficients, − compactly supported windows/filters are necessary.

Live demonstration



Sliding DGT spectrogram Sliding CQT spectrogram

Other demos: sliding Erblets, vocoder effects, denoising, DGT multiplier, parametric equalizer.

References

[1] P. L. Søndergaard, B. Torrésani, P. Balazs. *The Linear Time-Frequency Analysis Toolbox*. International Journal of Wavelets, Multiresolution Analysis and Information Processing, 10(4), 2012.

[2] Z. Průša, P. L. Søndergaard, N. Holighaus, Ch. Wiesmeyr, P. Balazs. *The Large Time-Frequency Analysis Toolbox 2.0*. (preprint), 2014, URL: `http://ltfat.sourceforge.net/notes/ltfatnote030.pdf`

[3] T. Necciari, P. Balazs, N. Holighaus, and P. L. Søndergaard. The ERBlet transform: An auditory-based time-frequency representation with perfect reconstruction. *In Proceedings of ICASSP 2013*, pages 498-502

Acknowledgment: Work presented in this poster was supported by the Austrian Science Fund (FWF) START-project FLAME ("Frames and Linear Operators for Acoustical Modeling and Parameter Estimation"; Y 551-N13).