MDCT Documentation

Release 0.4

Nils Werner

January 16, 2017

Contents

This toolkit implements several related transforms and their inverses:

- **Modified Discrete Cosine Transform (MDCT)**
- Modified Discrete Sine Transform (MDST)
- Modulated Complex Lapped Transform (MCLT) aka Complex Modified Discrete Cosine Transform (CMDCT)

All transforms are implemented as

- the complete lapped transform, along with windowing and time domain aliasing cancellation (TDAC) reconstruction and
- the core un-windowed standalone transform.

All transforms are implemeted in

- *mdct.fast*,a fast, FFT-based method (for actual use), see [Bosi]
- `mdct.slow`, a slow, pure-Python fashion (for testing) and

CHAPTER 1

Usage

Warning: *mdct.fast* is exposed as *mdct*. Please use this module directly.

```
import mdct
spec = mdct.mdct(signal)
output = mdct.imdct(spec)
```

1.1 Modules

1.1.1 mdct module

mdct.**mdct**(*x*, *odd=True*, *transforms=None*, ***kwargs*)

Calculate lapped MDCT of input signal

Parameters

- **x** (*array_like*) – The signal to be transformed. May be a 1D vector for single channel or a 2D matrix for multi channel data. In case of a mono signal, the data is must be a 1D vector of length `samples`. In case of a multi channel signal, the data must be in the shape of `samples x channels`.
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.
- **framelength** (*int*) – The signal frame length. Defaults to `2048`.
- **hopsize** (*int*) – The signal frame hopsize. Defaults to `None`. Setting this value will override `overlap`.
- **overlap** (*int*) – The signal frame overlap coefficient. Value `x` means `1/x` overlap. Defaults to `2`. Note that anything but `2` will result in a filterbank without perfect reconstruction.
- **centered** (*boolean*) – Pad input signal so that the first and last window are centered around the beginning of the signal. Defaults to `True`. Disabling this will result in aliasing in the first and last half-frame.
- **window** (*callable, array_like*) – Window to be used for deringing. Can be `False` to disable windowing. Defaults to `scipy.signal.cosine`.
- **transforms** (*module, optional*) – Module reference to core transforms. Mostly used to replace fast with slow core transforms, for testing. Defaults to *mdct.fast*
- **padding** (*int*) – Zero-pad signal with x times the number of samples. Defaults to `0`.

- **save_settings** (*boolean*) – Save settings used here in attribute `out.stft_settings` so that `ispectrogram()` can infer these settings without the developer having to pass them again.

Returns **out** – The signal (or matrix of signals). In case of a mono output signal, the data is formatted as a 1D vector of length `samples`. In case of a multi channel output signal, the data is formatted as `samples x channels`.

Return type array_like

See also:

mdct.fast.transforms.mdct() MDCT

mdct.**imdct**(*X*, *odd=True*, *transforms=None*, ***kwargs*)

Calculate lapped inverse MDCT of input signal

Parameters

- **x** (*array_like*) – The spectrogram to be inverted. May be a 2D matrix for single channel or a 3D tensor for multi channel data. In case of a mono signal, the data must be in the shape of `bins x frames`. In case of a multi channel signal, the data must be in the shape of `bins x frames x channels`.
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.
- **framelength** (*int*) – The signal frame length. Defaults to infer from data.
- **hopsize** (*int*) – The signal frame hopsize. Defaults to infer from data. Setting this value will override `overlap`.
- **overlap** (*int*) – The signal frame overlap coefficient. Value `x` means `1/x` overlap. Defaults to infer from data. Note that anything but `2` will result in a filterbank without perfect reconstruction.
- **centered** (*boolean*) – Pad input signal so that the first and last window are centered around the beginning of the signal. Defaults to to infer from data. The first and last half-frame will have aliasing, so using centering during forward MDCT is recommended.
- **window** (*callable, array_like*) – Window to be used for deringing. Can be `False` to disable windowing. Defaults to to infer from data.
- **halved** (*boolean*) – Switch to reconstruct the other halve of the spectrum if the forward transform has been truncated. Defaults to to infer from data.
- **transforms** (*module, optional*) – Module reference to core transforms. Mostly used to replace fast with slow core transforms, for testing. Defaults to *mdct.fast*
- **padding** (*int*) – Zero-pad signal with x times the number of samples. Defaults to infer from data.
- **outlength** (*int*) – Crop output signal to length. Useful when input length of spectrogram did not fit into framelength and input data had to be padded. Not setting this value will disable cropping, the output data may be longer than expected.

Returns **out** – The output signal

Return type array_like

See also:

mdct.fast.transforms.imdct() inverse MDCT

mdct.**mdst**(*x*, *odd=True*, *transforms=None*, ***kwargs*)
Calculate lapped MDST of input signal

Parameters

- **x** (*array_like*) – The input signal
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.
- **transforms** (*module, optional*) – Module reference to core transforms. Mostly used to replace fast with slow core transforms, for testing. Defaults to *mdct.fast* Additional keyword arguments passed to `stft.spectrogram`

Returns **out** – The output signal

Return type array_like

See also:

mdct.fast.transforms.mdst() MDST

mdct.**imdst**(*X*, *odd=True*, *transforms=None*, ***kwargs*)
Calculate lapped inverse MDST of input signal

Parameters

- **x** (*array_like*) – The input signal
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.
- **transforms** (*module, optional*) – Module reference to core transforms. Mostly used to replace fast with slow core transforms, for testing. Defaults to *mdct.fast* Additional keyword arguments passed to `stft.spectrogram`

Returns **out** – The output signal

Return type array_like

See also:

mdct.fast.transforms.imdst() inverse MDST

mdct.**cmdct**(*x*, *odd=True*, *transforms=None*, ***kwargs*)
Calculate lapped complex MDCT/MCLT of input signal

Parameters

- **x** (*array_like*) – The input signal
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.
- **transforms** (*module, optional*) – Module reference to core transforms. Mostly used to replace fast with slow core transforms, for testing. Defaults to *mdct.fast*
- **optional** (***kwargs,*) – Additional keyword arguments passed to `stft.spectrogram`

Returns **out** – The output signal

Return type array_like

See also:

mdct.fast.transforms.cmdct() complex MDCT

mdct.**icmdct**(*X*, *odd=True*, *transforms=None*, ***kwargs*)
Calculate lapped inverse complex MDCT/MCLT of input signal

Parameters

- **x** (*array_like*) – The input signal
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.
- **transforms** (*module, optional*) – Module reference to core transforms. Mostly used to replace fast with slow core transforms, for testing. Defaults to *mdct.fast* Additional keyword arguments passed to `stft.spectrogram`

Returns **out** – The output signal

Return type array_like

See also:

mdct.fast.transforms.icmdct() inverse complex MDCT

mdct.**mclt**(*x*, *odd=True*, *transforms=None*, ***kwargs*)
Calculate lapped complex MDCT/MCLT of input signal

Parameters

- **x** (*array_like*) – The input signal
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.
- **transforms** (*module, optional*) – Module reference to core transforms. Mostly used to replace fast with slow core transforms, for testing. Defaults to *mdct.fast*
- **optional** (***kwargs,*) – Additional keyword arguments passed to `stft.spectrogram`

Returns **out** – The output signal

Return type array_like

See also:

mdct.fast.transforms.cmdct() complex MDCT

mdct.**imclt**(*X*, *odd=True*, *transforms=None*, ***kwargs*)
Calculate lapped inverse complex MDCT/MCLT of input signal

Parameters

- **x** (*array_like*) – The input signal
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.
- **transforms** (*module, optional*) – Module reference to core transforms. Mostly used to replace fast with slow core transforms, for testing. Defaults to *mdct.fast* Additional keyword arguments passed to `stft.spectrogram`

Returns **out** – The output signal

Return type array_like

See also:

mdct.fast.transforms.icmdct() inverse complex MDCT

1.1.2 mdct.windows module

Module for windowing functions not found in SciPy

mdct.windows.**kaiser_derived**(*M*, *beta*)
Return a Kaiser-Bessel derived window.

Parameters

- **M** (*int*) – Number of points in the output window. If zero or less, an empty array is returned.
- **beta** (*float*) – Kaiser-Bessel window shape parameter.

Returns **w** – The window, normalized to fulfil the Princen-Bradley condition.

Return type ndarray

Notes

This window is only defined for an even number of taps.

References

1.2 Internal Modules

Warning: All necessary functions are exposed as the *mdct* module itself, please do not use internal modules directly.

1.2.1 mdct.fast module

Module for calculating lapped MDCT using FFT

Warning: Functions defined in this module are exposed using the *mdct* module itself, please do not use this module directly.

mdct.fast.**mdct**(*x*, *odd=True*, *transforms=None*, ***kwargs*)
Calculate lapped MDCT of input signal

Parameters

- **x** (*array_like*) – The signal to be transformed. May be a 1D vector for single channel or a 2D matrix for multi channel data. In case of a mono signal, the data is must be a 1D vector of length `samples`. In case of a multi channel signal, the data must be in the shape of `samples x channels`.
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.
- **framelength** (*int*) – The signal frame length. Defaults to `2048`.
- **hopsize** (*int*) – The signal frame hopsize. Defaults to `None`. Setting this value will override `overlap`.
- **overlap** (*int*) – The signal frame overlap coefficient. Value `x` means `1/x` overlap. Defaults to `2`. Note that anything but `2` will result in a filterbank without perfect reconstruction.

- **centered** (*boolean*) – Pad input signal so that the first and last window are centered around the beginning of the signal. Defaults to `True`. Disabling this will result in aliasing in the first and last half-frame.
- **window** (*callable, array_like*) – Window to be used for deringing. Can be `False` to disable windowing. Defaults to `scipy.signal.cosine`.
- **transforms** (*module, optional*) – Module reference to core transforms. Mostly used to replace fast with slow core transforms, for testing. Defaults to *mdct.fast*
- **padding** (*int*) – Zero-pad signal with x times the number of samples. Defaults to `0`.
- **save_settings** (*boolean*) – Save settings used here in attribute `out.stft_settings` so that `ispectrogram()` can infer these settings without the developer having to pass them again.

Returns **out** – The signal (or matrix of signals). In case of a mono output signal, the data is formatted as a 1D vector of length `samples`. In case of a multi channel output signal, the data is formatted as `samples x channels`.

Return type array_like

See also:

mdct.fast.transforms.mdct() MDCT

mdct.fast.**imdct**(*X*, *odd=True*, *transforms=None*, ***kwargs*)

Calculate lapped inverse MDCT of input signal

Parameters

- **x** (*array_like*) – The spectrogram to be inverted. May be a 2D matrix for single channel or a 3D tensor for multi channel data. In case of a mono signal, the data must be in the shape of `bins x frames`. In case of a multi channel signal, the data must be in the shape of `bins x frames x channels`.
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.
- **framelength** (*int*) – The signal frame length. Defaults to infer from data.
- **hopsize** (*int*) – The signal frame hopsize. Defaults to infer from data. Setting this value will override `overlap`.
- **overlap** (*int*) – The signal frame overlap coefficient. Value `x` means `1/x` overlap. Defaults to infer from data. Note that anything but `2` will result in a filterbank without perfect reconstruction.
- **centered** (*boolean*) – Pad input signal so that the first and last window are centered around the beginning of the signal. Defaults to to infer from data. The first and last half-frame will have aliasing, so using centering during forward MDCT is recommended.
- **window** (*callable, array_like*) – Window to be used for deringing. Can be `False` to disable windowing. Defaults to to infer from data.
- **halved** (*boolean*) – Switch to reconstruct the other halve of the spectrum if the forward transform has been truncated. Defaults to to infer from data.
- **transforms** (*module, optional*) – Module reference to core transforms. Mostly used to replace fast with slow core transforms, for testing. Defaults to *mdct.fast*
- **padding** (*int*) – Zero-pad signal with x times the number of samples. Defaults to infer from data.

- **outlength** (*int*) – Crop output signal to length. Useful when input length of spectrogram did not fit into framelength and input data had to be padded. Not setting this value will disable cropping, the output data may be longer than expected.

Returns **out** – The output signal

Return type array_like

See also:

mdct.fast.transforms.imdct() inverse MDCT

mdct.fast.**mdst**(*x*, *odd=True*, *transforms=None*, ***kwargs*)
Calculate lapped MDST of input signal

Parameters

- **x** (*array_like*) – The input signal
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.
- **transforms** (*module, optional*) – Module reference to core transforms. Mostly used to replace fast with slow core transforms, for testing. Defaults to *mdct.fast* Additional keyword arguments passed to `stft.spectrogram`

Returns **out** – The output signal

Return type array_like

See also:

mdct.fast.transforms.mdst() MDST

mdct.fast.**imdst**(*X*, *odd=True*, *transforms=None*, ***kwargs*)
Calculate lapped inverse MDST of input signal

Parameters

- **x** (*array_like*) – The input signal
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.
- **transforms** (*module, optional*) – Module reference to core transforms. Mostly used to replace fast with slow core transforms, for testing. Defaults to *mdct.fast* Additional keyword arguments passed to `stft.spectrogram`

Returns **out** – The output signal

Return type array_like

See also:

mdct.fast.transforms.imdst() inverse MDST

mdct.fast.**cmdct**(*x*, *odd=True*, *transforms=None*, ***kwargs*)
Calculate lapped complex MDCT/MCLT of input signal

Parameters

- **x** (*array_like*) – The input signal
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.

- **transforms** (*module, optional*) – Module reference to core transforms. Mostly used to replace fast with slow core transforms, for testing. Defaults to *mdct.fast*
- **optional** (***kwargs,*) – Additional keyword arguments passed to `stft.spectrogram`

Returns **out** – The output signal

Return type array_like

See also:

mdct.fast.transforms.cmdct() complex MDCT

`mdct.fast.`**icmdct**(*X*, *odd=True*, *transforms=None*, ***kwargs*)
Calculate lapped inverse complex MDCT/MCLT of input signal

Parameters

- **x** (*array_like*) – The input signal
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.
- **transforms** (*module, optional*) – Module reference to core transforms. Mostly used to replace fast with slow core transforms, for testing. Defaults to *mdct.fast* Additional keyword arguments passed to `stft.spectrogram`

Returns **out** – The output signal

Return type array_like

See also:

mdct.fast.transforms.icmdct() inverse complex MDCT

`mdct.fast.`**mclt**(*x*, *odd=True*, *transforms=None*, ***kwargs*)
Calculate lapped complex MDCT/MCLT of input signal

Parameters

- **x** (*array_like*) – The input signal
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.
- **transforms** (*module, optional*) – Module reference to core transforms. Mostly used to replace fast with slow core transforms, for testing. Defaults to *mdct.fast*
- **optional** (***kwargs,*) – Additional keyword arguments passed to `stft.spectrogram`

Returns **out** – The output signal

Return type array_like

See also:

mdct.fast.transforms.cmdct() complex MDCT

`mdct.fast.`**imclt**(*X*, *odd=True*, *transforms=None*, ***kwargs*)
Calculate lapped inverse complex MDCT/MCLT of input signal

Parameters

- **x** (*array_like*) – The input signal

- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.
- **transforms** (*module, optional*) – Module reference to core transforms. Mostly used to replace fast with slow core transforms, for testing. Defaults to *mdct.fast* Additional keyword arguments passed to `stft.spectrogram`

Returns **out** – The output signal

Return type array_like

See also:

mdct.fast.transforms.icmdct() inverse complex MDCT

1.2.2 mdct.fast.transforms module

Module for calculating DCT type 4 using FFT and pre/post-twiddling

Warning: These core transforms will produce aliasing when used without overlap. Please use *mdct* unless you know what this means.

`mdct.fast.transforms.`**mdct**(*x*, *odd=True*)
Calculate modified discrete cosine transform of input signal

Parameters

- **x** (*array_like*) – The input signal
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.

Returns **out** – The output signal

Return type array_like

`mdct.fast.transforms.`**imdct**(*X*, *odd=True*)
Calculate inverse modified discrete cosine transform of input signal

Parameters

- **X** (*array_like*) – The input signal
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.

Returns **out** – The output signal

Return type array_like

`mdct.fast.transforms.`**mdst**(*x*, *odd=True*)
Calculate modified discrete sine transform of input signal

Parameters

- **x** (*array_like*) – The input signal
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.

Returns **out** – The output signal

Return type array_like

`mdct.fast.transforms.`**imdst**(*X*, *odd=True*)
Calculate inverse modified discrete sine transform of input signal

Parameters

- **X** (*array_like*) – The input signal
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.

Returns **out** – The output signal

Return type array_like

`mdct.fast.transforms.`**`cmdct`**(*x*, *odd=True*)
Calculate complex MDCT/MCLT of input signal

Parameters

- **x** (*array_like*) – The input signal
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.

Returns **out** – The output signal

Return type array_like

`mdct.fast.transforms.`**`icmdct`**(*X*, *odd=True*)
Calculate inverse complex MDCT/MCLT of input signal

Parameters

- **X** (*array_like*) – The input signal
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.

Returns **out** – The output signal

Return type array_like

`mdct.fast.transforms.`**`mclt`**(*x*, *odd=True*)
Calculate complex MDCT/MCLT of input signal

Parameters

- **x** (*array_like*) – The input signal
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.

Returns **out** – The output signal

Return type array_like

`mdct.fast.transforms.`**`imclt`**(*X*, *odd=True*)
Calculate inverse complex MDCT/MCLT of input signal

Parameters

- **X** (*array_like*) – The input signal
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.

Returns **out** – The output signal

Return type array_like

1.2.3 mdct.slow.transforms module

Module for calculating DCT type 4 using pure Python

Warning: These core transforms will produce aliasing when used without overlap. Please use *mdct* unless you know what this means.

mdct.slow.transforms.**mdct**(*x*, *odd=True*)
Calculate modified discrete cosine transform of input signal in an inefficient pure-Python method.

Use only for testing.

Parameters

- **x** (*array_like*) – The input signal
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.

Returns **out** – The output signal

Return type array_like

mdct.slow.transforms.**imdct**(*X*, *odd=True*)
Calculate inverse modified discrete cosine transform of input signal in an inefficient pure-Python method.

Use only for testing.

Parameters

- **x** (*array_like*) – The input signal
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.

Returns **out** – The output signal

Return type array_like

mdct.slow.transforms.**mdst**(*x*, *odd=True*)
Calculate modified discrete sine transform of input signal in an inefficient pure-Python method.

Use only for testing.

Parameters

- **x** (*array_like*) – The input signal
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.

Returns **out** – The output signal

Return type array_like

mdct.slow.transforms.**imdst**(*X*, *odd=True*)
Calculate inverse modified discrete sine transform of input signal in an inefficient pure-Python method.

Use only for testing.

Parameters

- **x** (*array_like*) – The input signal
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.

Returns **out** – The output signal

Return type array_like

mdct.slow.transforms.**cmdct**(*x*, *odd=True*)
Calculate complex modified discrete cosine transform of input inefficient pure-Python method.

Use only for testing.

Parameters

- **x** (*array_like*) – The input signal
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.

Returns **out** – The output signal

Return type array_like

mdct.slow.transforms.**icmdct**(*X*, *odd=True*)

Calculate inverse complex modified discrete cosine transform of input signal in an inefficient pure-Python method.

Use only for testing.

Parameters

- **X** (*array_like*) – The input signal
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.

Returns **out** – The output signal

Return type array_like

mdct.slow.transforms.**mclt**(*x*, *odd=True*)

Calculate complex modified discrete cosine transform of input inefficient pure-Python method.

Use only for testing.

Parameters

- **X** (*array_like*) – The input signal
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.

Returns **out** – The output signal

Return type array_like

mdct.slow.transforms.**imclt**(*X*, *odd=True*)

Calculate inverse complex modified discrete cosine transform of input signal in an inefficient pure-Python method.

Use only for testing.

Parameters

- **X** (*array_like*) – The input signal
- **odd** (*boolean, optional*) – Switch to oddly stacked transform. Defaults to `True`.

Returns **out** – The output signal

Return type array_like

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Bibliography

[R1] Wikipedia, “Kaiser window”, https://en.wikipedia.org/wiki/Kaiser_window

[Bosi] Marina Bosi, Richard E. Goldberg and Leonardo Chiariglione, “Introduction to Digital Audio Coding and Standards”, Kluwer Academic Publishers, 01 December, 2002.

m

C

I

K

M