pysptk Documentation

Release 0.1.2

Ryuichi YAMAMOTO

Aug 03, 2018

Contents

A python wrapper for Speech Signal Processing Toolkit (SPTK).

https://github.com/r9y9/pysptk

CHAPTER 1

Full documentation

A full documentation of SPTK is available at http://sp-tk.sourceforge.net.

The wrapper is based on a modified version of SPTK (r9y9/SPTK)

CHAPTER 2

Installation guide

2.1 Installation guide

The latest release is availabe on pypi. You can install it by:

```
pip install pysptk
```

Note that you have to install `numpy` to build C-extensions.

If yout want the latest development version, run:

```
pip install git+https://github.com/r9y9/pysptk
```

or:

```
git clone https://github.com/r9y9/pysptk
cd pysptk
python setup.py develop # or install
```

This should resolve the package dependencies and install `pysptk` property.

Note: If you use the development version, you need `cython` installed to compile cython module(s).

CHAPTER 3

API documentation

3.1 API

3.1.1 Core SPTK API

All functionality in `pysptk.sptk` is directly accesible from the top-level `pysptk.*` namespace.

Note: Almost all of pysptk functions assume that the input array is **C-contiguous** and has `float64` element type.

Library routines

agexp	Magnitude squared generalized exponential function
gexp	Generalized exponential function
glog	Generalized logarithmic function
mseq	M-sequence

pysptk.sptk.agexp

`pysptk.sptk.`**`agexp`**`()`

Magnitude squared generalized exponential function

Parameters

r [float] Gamma

x [float] Real part

y [float] Imaginary part

Returns

Value

pysptk.sptk.gexp

pysptk.sptk.**gexp**()
Generalized exponential function

Parameters

r [float] Gamma

x [float] Arg

Returns

Value

pysptk.sptk.glog

pysptk.sptk.**glog**()
Generalized logarithmic function

Parameters

r [float] Gamma

x [float] Arg

Returns

Value

pysptk.sptk.mseq

pysptk.sptk.**mseq**()
M-sequence

Returns

A sample of m-sequence

Adaptive cepstrum analysis

acep	Adaptive cepstral analysis
agcep	Adaptive generalized cepstral analysis
amcep	Adaptive mel-cepstral analysis

pysptk.sptk.acep

pysptk.sptk.**acep**()
Adaptive cepstral analysis

Parameters

x [double] A input sample

c [array, shape(`order + 1`)] Cepstrum. The result is stored in place.

lambda_coef [float, optional] Leakage factor. Default is 0.98.

step [float, optional] Step size. Default is 0.1.

tau [float, optional] Momentum constant. Default is 0.9.

pd [int, optional] Order of pade approximation. Default is 4.

eps [float, optional] Minimum value for epsilon. Default is 1.0e-6.

Returns

prederr [float] Prediction error

Raises

ValueError if invalid order of pade approximation is specified

See also:

pysptk.sptk.uels, *pysptk.sptk.gcep*, *pysptk.sptk.mcep*, *pysptk.sptk.mgcep*, *pysptk.sptk.amcep*, *pysptk.sptk.agcep*, *pysptk.sptk.lmadf*

pysptk.sptk.agcep

pysptk.sptk.**agcep**()

Adaptive generalized cepstral analysis

Parameters

x [float] A input sample

c [array, shape(`order + 1`), optional] Cepstrum. The result is stored in-place.

stage [int, optional] -1 / gamma. Default is 1.

lambda_coef [float, optional] Leakage factor. Default is 0.98.

step [float, optional] Step size. Default is 0.1.

tau [float, optional] Momentum constant. Default is 0.9.

eps [float, optional] Minimum value for epsilon. Default is 1.0e-6.

Returns

prederr [float] Prediction error

Raises

ValueError if invalid number of stage is specified

See also:

pysptk.sptk.acep, *pysptk.sptk.amcep*, *pysptk.sptk.glsadf*

pysptk.sptk.amcep

pysptk.sptk.**amcep**()

Adaptive mel-cepstral analysis

Parameters

x [float] A input sample

b [array, shape(`order + 1`), optional] MLSA filter coefficients. The result is stored in-place.

alpha [float, optional] All-pass constant. Default is 0.35.

lambda_coef [float, optional] Leakage factor. Default is 0.98.

step [float, optional] Step size. Default is 0.1.

tau [float, optional] Momentum constant. Default is 0.9.

pd [int, optional] Order of pade approximation. Default is 4.

eps [float, optional] Minimum value for epsilon. Default is 1.0e-6.

Returns

prederr [float] Prediction error

Raises

ValueError if invalid order of pade approximation is specified

See also:

pysptk.sptk.acep, *pysptk.sptk.agcep*, *pysptk.sptk.mc2b*, *pysptk.sptk.b2mc*, *pysptk.sptk.mlsadf*

Mel-generalized cepstrum analysis

mcep	Mel-cepstrum analysis
gcep	Generalized-cepstrum analysis
mgcep	Mel-generalized cepstrum analysis
uels	Unbiased estimation of log spectrum
fftcep	FFT-based cepstrum analysis
lpc	Linear prediction analysis

pysptk.sptk.mcep

pysptk.sptk.**mcep**()

Mel-cepstrum analysis

Parameters

windowed [array, shape (`frame_len`)] A windowed frame

order [int, optional] Order of mel-cepstrum. Default is 25.

alpha [float, optional] All pass constant. Default is 0.35.

miniter [int, optional] Minimum number of iteration. Default is 2.

maxiter [int, optional] Maximum number of iteration. Default is 30.

threshold [float, optional] Threshold in theq. Default is 0.001.

etype [int, optional]

Type of parameter `eps`

0. not used
1. initial value of log-periodogram

2. floor of periodogram in db

Default is 0.

eps [float, optional] Initial value for log-periodogram or floor of periodogram in db. Default is 0.0.

min_det [float, optional] Mimimum value of the determinant of normal matrix. Default is 1.0e-6

itype [float, optional]

Input data type

0. windowed signal
1. log amplitude in db
2. log amplitude
3. amplitude
4. periodogram

Default is 0.

Returns

mc [array, shape (`order + 1`)] Mel-cepstrum

Raises

ValueError

- if invalid `itype` is specified
- if invalid `etype` is specified
- if nonzero `eps` is specified when etype = 0
- if negative `eps` is specified
- if negative `min_det` is specified

RuntimeError

- if zero(s) are found in periodogram
- if error happened in theq

See also:

pysptk.sptk.uels, *pysptk.sptk.gcep*, *pysptk.sptk.mgcep*, *pysptk.sptk.mlsadf*

pysptk.sptk.gcep

pysptk.sptk.**gcep**()

Generalized-cepstrum analysis

Parameters

windowed [array, shape (`frame_len`)] A windowed frame

order [int, optional] Order of generalized-cepstrum. Default is 25.

gamma [float, optional] Parameter of generalized log function. Default is 0.0.

miniter [int, optional] Minimum number of iteration. Default is 2.

maxiter [int, optional] Maximum number of iteration. Default is 30.

threshold [float, optional] Threshold in theq. Default is 0.001

etype [int, optional]

Type of parameter `eps`

0. not used
1. initial value of log-periodogram
2. floor of periodogram in db

Default is 0.

eps [float, optional] Initial value for log-periodogram or floor of periodogram in db. Default is 0.0.

min_det [float, optional] Mimimum value of the determinant of normal matrix. Default is 1.0e-6.

itype [float, optional]

Input data type

0. windowed signal
1. log amplitude in db
2. log amplitude
3. amplitude
4. periodogram

Default is 0.

Returns

gc [array, shape (`order + 1`)] Generalized cepstrum

Raises

ValueError

- if invalid `itype` is specified
- if invalid `etype` is specified
- if nonzero `eps` is specified when etype = 0
- if negative `eps` is specified
- if negative `min_det` is specified

RuntimeError

- if error happened in theq

See also:

pysptk.sptk.uels, *pysptk.sptk.mcep*, *pysptk.sptk.mgcep*, *pysptk.sptk.glsadf*

pysptk.sptk.mgcep

pysptk.sptk.**mgcep**()

Mel-generalized cepstrum analysis

Parameters

windowed [array, shape (`frame_len`)] A windowed frame

order [int, optional] Order of mel-generalized cepstrum. Default is 25.

alpha [float, optional] All pass constant. Default is 0.35.

gamma [float, optional] Parameter of generalized log function. Default is 0.0.

num_recursions [int, optional] Number of recursions. Default is `len(windowed) - 1`.

miniter [int, optional] Minimum number of iteration. Default is 2.

maxiter [int, optional] Maximum number of iteration. Default is 30.

threshold [float, optional] Threshold. Default is 0.001.

etype [int, optional]

Type of paramter `e`

0. not used
1. initial value of log-periodogram
2. floor of periodogram in db

Default is 0.

eps [float, optional] Initial value for log-periodogram or floor of periodogram in db. Default is 0.0.

min_det [float, optional] Mimimum value of the determinant of normal matrix. Default is 1.0e-6.

itype [float, optional]

Input data type

0. windowed signal
1. log amplitude in db
2. log amplitude
3. amplitude
4. periodogram

Default is 0.

otype [int, optional]

Output data type

0. mel generalized cepstrum: (c~0...c~m)
1. MGLSA filter coefficients: b0...bm
2. K~,c~'1...c~'m
3. K,b'1...b'm
4. K~,g*c~'1...g*c~'m

5. K,g*b‘1...g*b’m

Default is 0.

Returns

mgc [array, shape (`order + 1`)] mel-generalized cepstrum

Raises

ValueError

- if invalid `itype` is specified
- if invalid `etype` is specified
- if nonzero `eps` is specified when etype = 0
- if negative `eps` is specified
- if negative `min_det` is specified
- if invalid `otype` is specified

RuntimeError

- if error happened in theq

See also:

pysptk.sptk.uels, *pysptk.sptk.gcep*, *pysptk.sptk.mcep*, *pysptk.sptk.freqt*, *pysptk.sptk.gc2gc*, *pysptk.sptk.mgc2mgc*, *pysptk.sptk.gnorm*, *pysptk.sptk.mglsadf*

pysptk.sptk.uels

pysptk.sptk.**uels**()

Unbiased estimation of log spectrum

Parameters

windowed [array, shape (`frame_len`)] A windowed frame

order [int, optional] Order of cepstrum. Default is 25.

miniter [int, optional] Minimum number of iteration. Default is 2.

maxiter [int, optional] Maximum number of iteration. Default is 30.

threshold [float, optional] Threshold in theq. Default is 0.001

etype [int, optional]

Type of parameter `eps`

0. not used
1. initial value of log-periodogram
2. floor of periodogram in db

Default is 0.

eps [float, optional] Initial value for log-periodogram or floor of periodogram in db. Default is 0.0.

itype [float, optional]

Input data type

0. windowed signal
1. log amplitude in db
2. log amplitude
3. amplitude
4. periodogram

Default is 0.

Returns

c [array, shape (`order + 1`)] cepstrum estimated by uels

Raises

ValueError

- if invalid `itype` is specified
- if invalid `etype` is specified
- if nonzero `eps` is specified when etype = 0
- if negative `eps` is specified

RuntimeError

- if zero(s) are found in periodogram

See also:

pysptk.sptk.gcep, *pysptk.sptk.mcep*, *pysptk.sptk.mgcep*, *pysptk.sptk.lmadf*

pysptk.sptk.fftcep

pysptk.sptk.**fftcep**()

FFT-based cepstrum analysis

Parameters

logsp [array, shape (`frame_len`)] Log power spectrum

order [int, optional] Order of cepstrum. Default is 25.

num_iter [int, optional] Number of iteration. Default is 0.

acceleration_factor [float, optional] Acceleration factor. Default is 0.0.

Returns

c [array, shape (`order + 1`)] Cepstrum

See also:

pysptk.sptk.uels

pysptk.sptk.lpc

pysptk.sptk.**lpc**()
Linear prediction analysis

Parameters

windowed [array, shape (`frame_len`)] A windowed frame

order [int, optional] Order of LPC. Default is 25.

min_det [float, optional] Mimimum value of the determinant of normal matrix. Default is 1.0e-6.

Returns

a [array, shape (`order + 1`)] LPC

Raises

ValueError

- if negative `min_det` is specified

RuntimeError

- if error happened in levdur

See also:

pysptk.sptk.lpc2par, *pysptk.sptk.par2lpc*, *pysptk.sptk.lpc2c*, *pysptk.sptk.lpc2lsp*, *pysptk.sptk.ltcdf*, *pysptk.sptk.lspdf*

MFCC

mfcc	MFCC

pysptk.sptk.mfcc

pysptk.sptk.**mfcc**()
MFCC

Parameters

x [array] A input signal

order [int, optional] Order of MFCC. Default is 14.

fs [int, optional] Sampling frequency. Default is 160000.

alpha [float, optional] Pre-emphasis coefficient. Default is 0.97.

eps [float, optional] Flooring value for calculating `log(x)` in filterbank analysis. Default is 1.0.

window_len [int, optional] Window lenght. Default is `len(x)`.

frame_len [int, optional] Frame length. Default is `len(x)`.

num_filterbanks [int, optional] Number of mel-filter banks. Default is 20.

cepslift [int, optional] Liftering coefficient. Default is 22.

use_dft [bool, optional] Use DFT (not FFT) or not. Default is False.

use_hamming [bool, optional] Use hamming window or not. Default is False.

czero [bool, optional] If True, `mfcc` returns 0-th coefficient as well. Default is False.

power [bool, optional] If True, `mfcc` returns power coefficient as well. Default is False.

Returns

cc [array] MFCC vector, which is ordered as:

mfcc[0], mfcc[1], mfcc[2], ... mfcc[order-1], c0, Power.

Note that c0 and Power are optional.

Shape of `cc` is:

- `order` by default.
- `orde + 1` if `czero` or `power` is set to True.
- `order + 2` if both `czero` and `power` is set to True.

Raises

ValueError if `num_filterbanks` is less than or equal to `order`

See also:

pysptk.sptk.gcep, *pysptk.sptk.mcep*, *pysptk.sptk.mgcep*

LPC, LSP and PARCOR conversions

lpc2c	LPC to cepstrum
lpc2lsp	LPC to LSP
lpc2par	LPC to PARCOR
par2lpc	PARCOR to LPC
lsp2sp	LSP to spectrum

pysptk.sptk.lpc2c

pysptk.sptk.**lpc2c**()

LPC to cepstrum

Parameters

lpc [array] LPC

order [int, optional] Order of cepstrum. Default is `len(lpc) - 1`.

Returns

ceps [array, shape (`order + 1`)] cepstrum

See also:

pysptk.sptk.lpc, *pysptk.sptk.lspdf*

pysptk.sptk.lpc2lsp

pysptk.sptk.**lpc2lsp**()
LPC to LSP

Parameters

lpc [array] LPC

numsp [int, optional] Number of unit circle. Default is 512.

maxiter [int, optional] Maximum number of iteration. Default is 4.

eps [float, optional] End condition for iteration. Default is 1.0e-6.

loggain [bool, optional] whether the converted lsp should have loggain or not. Default is False.

fs [int, optional] Sampling frequency. Default is None and unused.

otype [int, optional]

Output format LSP

0. normalized frequency (0 ~ pi)
1. normalized frequency (0 ~ 0.5)
2. frequency (kHz)
3. frequency (Hz)

Default is 0.

Returns

lsp [array, shape (`order + 1`)] LSP

Raises

ValueError if `fs` is not specified when otype = 2 or 3.

See also:

pysptk.sptk.lpc, *pysptk.sptk.lspdf*

pysptk.sptk.lpc2par

pysptk.sptk.**lpc2par**()
LPC to PARCOR

Parameters

lpc [array] LPC

Returns

par [array, shape (same as `lpc`)] PARCOR

See also:

pysptk.sptk.lpc, *pysptk.sptk.par2lpc*, *pysptk.sptk.ltcdf*

pysptk.sptk.par2lpc

pysptk.sptk.**par2lpc**()
PARCOR to LPC

Parameters

par [array] PARCOR

Returns

lpc [array, shape (same as `par`)] LPC

See also:

pysptk.sptk.lpc, *pysptk.sptk.lpc2par*

pysptk.sptk.lsp2sp

pysptk.sptk.**lsp2sp**()
LSP to spectrum

Parameters

lsp [array] LSP

fftlen [int, optional] FFT length

TODO: consider ‘‘otype‘‘ optional argument

Returns

sp [array, shape] Spectrum. ln|H(z)|.

See also:

pysptk.sptk.lpc2par

Notes

It is asuumed that `lsp` has loggain at `lsp[0]`.

Mel-generalized cepstrum conversions

mc2b	Mel-cepsrum to MLSA filter coefficients
b2mc	MLSA filter coefficients to mel-cesptrum
c2acr	Cepstrum to autocorrelation
c2ir	Cepstrum to impulse response
ic2ir	Impulse response to cepstrum
c2ndps	Cepstrum to Negative Derivative of Phase Spectrum (NDPS)
ndps2c	Cepstrum to Negative Derivative of Phase Spectrum (NDPS)
gc2gc	Generalized cepstrum transform
gnorm	Gain normalization
ignorm	Inverse gain normalization

Continued on next page

Table 6 – continued from previous page

freqt	Frequency transform
mgc2mgc	Mel-generalized cepstrum transform
mgc2sp	Mel-generalized cepstrum transform
mgclsp2sp	MGC-LSP to spectrum

pysptk.sptk.mc2b

pysptk.sptk.**mc2b**()
Mel-cepsrum to MLSA filter coefficients

Parameters

mc [array, shape] Mel-cepstrum.

alpha [float, optional] All-pass constant. Default is 0.35.

Returns

b [array, shape(same as `mc`)] MLSA filter coefficients

See also:

pysptk.sptk.mlsadf, *pysptk.sptk.mglsadf*, *pysptk.sptk.b2mc*, *pysptk.sptk.mcep*, *pysptk.sptk.mgcep*, *pysptk.sptk.amcep*

pysptk.sptk.b2mc

pysptk.sptk.**b2mc**()
MLSA filter coefficients to mel-cesptrum

Parameters

b [array, shape] MLSA filter coefficients

alpha [float, optional] All-pass constant. Default is 0.35.

Returns

mc [array, shape (same as `b`)] Mel-cepstrum.

See also:

pysptk.sptk.mc2b, *pysptk.sptk.mcep*, *pysptk.sptk.mlsadf*

pysptk.sptk.c2acr

pysptk.sptk.**c2acr**()
Cepstrum to autocorrelation

Parameters

c [array] Cepstrum

order [int, optional] Order of cepstrum. Default is `len(c) - 1`.

fftlen [int, optional] FFT length. Default is 256.

Returns

r [array, shape (`order + 1`)] Autocorrelation

Raises

ValueError if non power of 2 `fftlen` is specified

See also:

pysptk.sptk.uels, *pysptk.sptk.c2ir*, *pysptk.sptk.lpc2c*

pysptk.sptk.c2ir

pysptk.sptk.**c2ir**()

Cepstrum to impulse response

Parameters

c [array] Cepstrum

length [int, optional] Length of impulse response. Default is 256.

Returns

h [array, shape (`length`)] impulse response

See also:

pysptk.sptk.c2acr

pysptk.sptk.ic2ir

pysptk.sptk.**ic2ir**()

Impulse response to cepstrum

Parameters

h [array] Impulse response

order [int, optional] Order of cepstrum. Default is 25.

Returns

c [array, shape (`order + 1`)] Cepstrum

See also:

pysptk.sptk.c2ir

pysptk.sptk.c2ndps

pysptk.sptk.**c2ndps**()

Cepstrum to Negative Derivative of Phase Spectrum (NDPS)

Parameters

c [array] Cepstrum

fftlen [int, optional] FFT length. Default is 256.

Returns

ndps [array, shape (`fftlen // 2 + 1`)] NDPS

Raises

ValueError if non power of 2 `fftlen` is specified

See also:

pysptk.sptk.mgcep, *pysptk.sptk.ndps2c*

pysptk.sptk.ndps2c

pysptk.sptk.**ndps2c**()
Cepstrum to Negative Derivative of Phase Spectrum (NDPS)

Parameters

ndps [array, shape (`fftlen // 2 + 1`)] NDPS

order [int, optional] Order of cepstrum. Default is 25.

Returns

c [array, shape (`order + 1`)] Cepstrum

Raises

ValueError if non power of 2 `fftlen` is detected

See also:

pysptk.sptk.mgc2sp, *pysptk.sptk.c2ndps*

pysptk.sptk.gc2gc

pysptk.sptk.**gc2gc**()
Generalized cepstrum transform

Parameters

src_ceps [array] Generalized cepstrum.

src_gamma [float, optional] Gamma of source cepstrum. Default is 0.0.

dst_order [int, optional] Order of destination cepstrum. Default is `len(src_ceps) - 1`.

dst_gamma [float, optional] Gamma of destination cepstrum. Default is 0.0.

Returns

dst_ceps [array, shape (`dst_order + 1`)] Converted generalized cepstrum

Raises

ValueError

- if invalid `src_gamma` is specified
- if invalid `dst_gamma` is specified

See also:

pysptk.sptk.gcep, *pysptk.sptk.mgcep*, *pysptk.sptk.freqt*, *pysptk.sptk.mgc2mgc*, *pysptk.sptk.lpc2c*

pysptk.sptk.gnorm

pysptk.sptk.**gnorm**()
Gain normalization

Parameters

ceps [array] Generalized cepstrum.

gamma [float, optional] Gamma. Default is 0.0.

Returns

dst_ceps [array, shape(same as ceps)] Normalized generalized cepstrum

Raises

ValueError if invalid gamma is specified

See also:

pysptk.sptk.ignorm, *pysptk.sptk.gcep*, *pysptk.sptk.mgcep*, *pysptk.sptk.gc2gc*, *pysptk.sptk.mgc2mgc*, *pysptk.sptk.freqt*

pysptk.sptk.ignorm

pysptk.sptk.**ignorm**()
Inverse gain normalization

Parameters

c [array] Normalized generalized cepstrum

gamma [float, optional] Gamma. Default is 0.0.

Returns

dst_ceps [array, shape (same as ceps)] Generalized cepstrum

Raises

ValueError if invalid gamma is specified

See also:

pysptk.sptk.gnorm, *pysptk.sptk.gcep*, *pysptk.sptk.mgcep*, *pysptk.sptk.gc2gc*, *pysptk.sptk.mgc2mgc*, *pysptk.sptk.freqt*

pysptk.sptk.freqt

pysptk.sptk.**freqt**()
Frequency transform

Parameters

ceps [array] Cepstrum.

order [int, optional] Desired order of transformed cepstrum. Default is 25.

alpha [float, optional] All-pass constant. Default is 0.0.

Returns

dst_ceps [array, shape(`order + 1`)] frequency transofmed cepsttrum (typically mel-cepstrum)

See also:

pysptk.sptk.mgc2mgc

pysptk.sptk.mgc2mgc

pysptk.sptk.**mgc2mgc**()
Mel-generalized cepstrum transform

Parameters

src_ceps [array] Mel-generalized cepstrum.

src_alpha [float, optional] All-pass constant of source cesptrum. Default is 0.0.

src_gamma [float, optional] Gamma of source cepstrum. Default is 0.0.

dst_order [int, optional] Order of destination cepstrum. Default is `len(src_ceps) - 1`.

dst_alpha [float, optional] All-pass constant of destination cesptrum. Default is 0.0.

dst_gamma [float, optional] Gamma of destination cepstrum. Default is 0.0.

Returns

dst_ceps [array, shape (`dst_order + 1`)] Converted mel-generalized cepstrum

Raises

ValueError

- if invalid `src_gamma` is specified
- if invalid `dst_gamma` is specified

See also:

pysptk.sptk.uels, *pysptk.sptk.gcep*, *pysptk.sptk.mcep*, *pysptk.sptk.mgcep*, *pysptk.sptk.gc2gc*, *pysptk.sptk.freqt*, *pysptk.sptk.lpc2c*

pysptk.sptk.mgc2sp

pysptk.sptk.**mgc2sp**()
Mel-generalized cepstrum transform

Parameters

ceps [array] Mel-generalized cepstrum.

alpha [float, optional] All-pass constant. Default is 0.0.

gamma [float, optional] Gamma. Default is 0.0.

fftlen [int, optional] FFT length. Default is 256.

Returns

sp [array, shape (`fftlen // 2 + 1`)] Complex spectrum

Raises

ValueError

- if invalid `gamma` is specified
- if non power of 2 `fftlen` is specified

See also:

pysptk.sptk.mgc2mgc, *pysptk.sptk.gc2gc*, *pysptk.sptk.freqt*, *pysptk.sptk.gnorm*, *pysptk.sptk.lpc2c*

pysptk.sptk.mgclsp2sp

pysptk.sptk.**mgclsp2sp**()
MGC-LSP to spectrum

Parameters

lsp [array] MGC-LSP

alpha [float, optional] All-pass constant. Default is 0.0.

gamma [float, optional] Gamma. Default is 0.0.

fftlen [int, optional] FFT length. Default is 256.

gain [bool, optional] Whether the input MGC-LSP should have loggain or not. Default is True.

Returns

sp [array, shape (`fftlen // 2 + 1`)] Complex spectrum

Raises

ValueError

- if invalid `gamma` is specified
- if non power of 2 `fftlen` is specified

See also:

pysptk.sptk.mgc2mgc

F0 analysis

swipe	SWIPE' - A Saw-tooth Waveform Inspired Pitch Estimation

pysptk.sptk.swipe

pysptk.sptk.**swipe**()
SWIPE' - A Saw-tooth Waveform Inspired Pitch Estimation

Parameters

x [array] A whole audio signal

fs [int] Sampling frequency.

hopsize [int] Hop size.

min [float, optional] Minimum fundamental frequency. Default is 50.0

max [float, optional] Maximum fundamental frequency. Default is 800.0

threshold [float, optional] Voice/unvoiced threshold. Default is 0.3.

otype [int, optional] Output format (0) pitch (1) f0 (2) log(f0). Default is 1.

Returns

f0 [array, shape(`len(x)/frame_shift+1`)] Estimated f0 trajectory

Raises

ValueError if invalid otype is specified

Examples

Window functions

blackman	Blackman window
hamming	Hamming window
hanning	Hanning window
bartlett	Bartlett window
trapezoid	Trapezoid window
rectangular	Rectangular window

pysptk.sptk.blackman

pysptk.sptk.**blackman**()
Blackman window

Parameters

n [int] Window length

normalize [int, optional]

Normalization flag

0. don't normalize
1. normalize by power
2. normalize by magnitude

Defalt is 0.

Returns

w [array, shape (n,)] blackman window

pysptk.sptk.hamming

pysptk.sptk.**hamming**()
Hamming window

Parameters

n [int] Window length

normalize [int, optional]

Normalization flag

0. don't normalize
1. normalize by power
2. normalize by magnitude

Defalt is 0.

Returns

w [array, shape (n,)] hamming window

pysptk.sptk.hanning

pysptk.sptk.**hanning**()

Hanning window

Parameters

n [int] Window length

normalize [int, optional]

Normalization flag

0. don't normalize
1. normalize by power
2. normalize by magnitude

Defalt is 0.

Returns

w [array, shape (n,)] hanning window

pysptk.sptk.bartlett

pysptk.sptk.**bartlett**()

Bartlett window

Parameters

n [int] Window length

normalize [int, optional]

Normalization flag

0. don't normalize
1. normalize by power
2. normalize by magnitude

Defalt is 0.

Returns

w [array, shape (n,)] bartlett window

pysptk.sptk.trapezoid

pysptk.sptk.**trapezoid**()
Trapezoid window

Parameters

n [int] Window length

normalize [int, optional]

Normalization flag

0. don't normalize
1. normalize by power
2. normalize by magnitude

Defalt is 0.

Returns

w [array, shape (n,)] trapezoid window

pysptk.sptk.rectangular

pysptk.sptk.**rectangular**()
Rectangular window

Parameters

n [int] Window length

normalize [int, optional]

Normalization flag

0. don't normalize
1. normalize by power
2. normalize by magnitude

Defalt is 0.

Returns

w [array, shape (n,)] rectangular window

Waveform generation filters

poledf	All-pole digital filter
lmadf	LMA digital filter
lspdf	LSP synthesis digital filter
ltcdf	All-pole lattice digital filter
glsadf	GLSA digital filter
mlsadf	MLSA digital filter
mglsadf	MGLSA digital filter

pysptk.sptk.poledf

pysptk.sptk.**poledf**()
All-pole digital filter

Parameters

x [float] A input sample

a [array] AR coefficients

delay [array] Delay

Returns

y [float] A filtered sample

Raises

ValueError if invalid delay length is supplied

See also:

pysptk.sptk.lpc, *pysptk.sptk.ltcdf*, *pysptk.sptk.lmadf*

pysptk.sptk.lmadf

pysptk.sptk.**lmadf**()
LMA digital filter

Parameters

x [float] A input sample

c [array] Cepstrum

pd [int] Order of pade approximation

delay [array] Delay

Returns

y [float] A filtered sample

Raises

ValueError

- if invalid order of pade approximation is specified
- if invalid delay length is supplied

See also:

pysptk.sptk.uels, *pysptk.sptk.acep*, *pysptk.sptk.poledf*, *pysptk.sptk.ltcdf*, *pysptk.sptk.glsadf*, *pysptk.sptk.mlsadf*, *pysptk.sptk.mglsadf*

pysptk.sptk.lspdf

pysptk.sptk.**lspdf**()
LSP synthesis digital filter

Parameters

x [float] A input sample

f [array] LSP coefficients

delay [array] Delay

Returns

y [float] A filtered sample

Raises

ValueError if invalid delay length is supplied

See also:

pysptk.sptk.lpc2lsp

pysptk.sptk.ltcdf

pysptk.sptk.**ltcdf**()
All-pole lattice digital filter

Parameters

x [float] A input sample

k [array] PARCOR coefficients.

delay [array] Delay

Returns

y [float] A filtered sample

Raises

ValueError if invalid delay length is supplied

See also:

pysptk.sptk.lpc, *pysptk.sptk.lpc2par*, *pysptk.sptk.lpc2lsp*, *pysptk.sptk.poledf*, *pysptk.sptk.lspdf*

pysptk.sptk.glsadf

pysptk.sptk.**glsadf**()
GLSA digital filter

Parameters

x [float] A input sample

c [array] Geneeraized cepstrum

stage [int] -1 / gamma

delay [array] Delay

Returns

y [float] A filtered sample

Raises

ValueError

- if invalid number of stage is specified
- if invalid delay length is supplied

See also:

pysptk.sptk.ltcdf, *pysptk.sptk.lmadf*, *pysptk.sptk.lspdf*, *pysptk.sptk.mlsadf*, *pysptk.sptk.mglsadf*

pysptk.sptk.mlsadf

pysptk.sptk.**mlsadf**()

MLSA digital filter

Parameters

x [float] A input sample

b [array] MLSA filter coefficients

alpha [float] All-pass constant

pd [int] Order of pade approximation

delay [array] Delay

Returns

y [float] A filtered sample

Raises

ValueError

- if invalid order of pade approximation is specified
- if invalid delay length is supplied

See also:

pysptk.sptk.mcep, *pysptk.sptk.amcep*, *pysptk.sptk.poledf*, *pysptk.sptk.ltcdf*, *pysptk.sptk.lmadf*, *pysptk.sptk.lspdf*, *pysptk.sptk.glsadf*, *pysptk.sptk.mglsadf*

pysptk.sptk.mglsadf

pysptk.sptk.**mglsadf**()

MGLSA digital filter

Parameters

x [float] A input sample

b [array] MGLSA filter coefficients

alpha [float] All-pass constant

stage [int] -1 / gamma

delay [array] Delay

Returns

y [float] A filtered sample

Raises

ValueError

- if invalid number of stage is specified
- if invalid delay length is supplied

See also:

pysptk.sptk.mgcep, *pysptk.sptk.poledf*, *pysptk.sptk.ltcdf*, *pysptk.sptk.lmadf*, *pysptk.sptk.lspdf*, *pysptk.sptk.mlsadf*, *pysptk.sptk.glsadf*

Utilities for waveform generation filters

poledf_delay	Delay for poledf
lmadf_delay	Delay for lmadf
lspdf_delay	Delay for lspdf
ltcdf_delay	Delay for ltcdf
glsadf_delay	Delay for glsadf
mlsadf_delay	Delay for mlsadf
mglsadf_delay	Delay for mglsadf

pysptk.sptk.poledf_delay

pysptk.sptk.**poledf_delay**()
Delay for poledf

Parameters

order [int] Order of poledf filter coefficients

Returns

delay [array] Delay

pysptk.sptk.lmadf_delay

pysptk.sptk.**lmadf_delay**()
Delay for lmadf

Parameters

order [int] Order of lmadf filter coefficients

pd [int] Order of pade approximation.

Returns

delay [array] Delay

pysptk.sptk.lspdf_delay

pysptk.sptk.**lspdf_delay**()
Delay for lspdf

Parameters

order [int] Order of lspdf filter coefficients

Returns

delay [array] Delay

pysptk.sptk.ltcdf_delay

pysptk.sptk.**ltcdf_delay**()
Delay for ltcdf

Parameters

order [int] Order of ltcdf filter coefficients

Returns

delay [array] Delay

pysptk.sptk.glsadf_delay

pysptk.sptk.**glsadf_delay**()
Delay for glsadf

Parameters

order [int] Order of glsadf filter coefficients

stage [int] -1 / gamma

Returns

delay [array] Delay

pysptk.sptk.mlsadf_delay

pysptk.sptk.**mlsadf_delay**()
Delay for mlsadf

Parameters

order [int] Order of mlsadf filter coefficients

pd [int] Order of pade approximation.

Returns

delay [array] Delay

pysptk.sptk.mglsadf_delay

pysptk.sptk.**mglsadf_delay**()
Delay for mglsadf

Parameters

order [int] Order of mglsadf filter coefficients

stage [int] -1 / gamma

Returns

delay [array] Delay

Other conversions

mgc2b(mgc[, alpha, gamma])	Mel-generalized cepstrum to MGLSA filter coefficients

pysptk.conversion.mgc2b

pysptk.conversion.**mgc2b**(*mgc*, *alpha=0.35*, *gamma=0.0*)

Mel-generalized cepstrum to MGLSA filter coefficients

Parameters

mgc [array, shape] Mel-generalized cepstrum

alpha [float] All-pass constant. Default is 0.35.

gamma [float] Parameter of generalized log function. Default is 0.0.

Returns

b [array, shape(same as mgc)] MGLSA filter coefficients

See also:

pysptk.sptk.mlsadf, *pysptk.sptk.mglsadf*, *pysptk.sptk.mc2b*, *pysptk.sptk.b2mc*, *pysptk.sptk.mcep*, *pysptk.sptk.mgcep*, *pysptk.sptk.mgcep*

3.1.2 High-level interface for waveform synthesis

Module pysptk.synthesis provides high-leve interface that wraps low-level SPTK waveform synthesis functions (e.g. mlsadf),

Synthesizer

class pysptk.synthesis.**Synthesizer**(*filt*, *hopsize*)

Speech waveform synthesizer

Attributes

filt [SynthesisFilter] A speech synthesis filter

hopsize [int] Hop size

synthesis(*source*, *b*)

Synthesize a waveform given a source excitation and sequence of filter coefficients (e.g. cepstrum).

Parameters

source [array] Source excitation

b [array] Filter coefficients

Returns

y [array, shape (same as source)] Synthesized waveform

synthesis_one_frame(*source*, *prev_b*, *curr_b*)
Synthesize one frame waveform

Parameters

source [array] Source excitation

prev_b [array] Filter coefficients of previous frame

curr_b [array] Filter coefficients of current frame

Returns

y [array] Synthesized waveform

SynthesisFilters

LMADF

class pysptk.synthesis.**LMADF**(*order=25*, *pd=4*)
LMA digital filter that wraps `lmadf`

Attributes

pd [int] Order of pade approximation. Default is 4.

delay [array] Delay

filt(*x*, *coef*)
Filter one sample using using `lmadf`

Parameters

x [float] A input sample

coef: array LMA filter coefficients (i.e. Cepstrum)

Returns

y [float] A filtered sample

See also:

pysptk.sptk.lmadf

MLSADF

class pysptk.synthesis.**MLSADF**(*order=25*, *alpha=0.35*, *pd=4*)
MLSA digital filter that wraps `mlsadf`

Attributes

alpha [float] All-pass constant

pd [int] Order of pade approximation. Default is 4.

delay [array] Delay

filt(*x*, *coef*)
Filter one sample using `mlsadf`

Parameters

x [float] A input sample

coef: array MLSA filter coefficients

Returns

y [float] A filtered sample

See also:

pysptk.sptk.mlsadf, *pysptk.sptk.mc2b*

MGLSADF

class pysptk.synthesis.**MGLSADF**(*order=25*, *alpha=0.35*, *stage=1*)

MGLSA digital filter that wraps `mglsadf`

Attributes

alpha [float] All-pass constant

stage [int] -1/gamma

delay [array] Delay

filt(*x*, *coef*)

Filter one sample using `mglsadf`

Parameters

x [float] A input sample

coef: array MGLSA filter coefficients

Returns

y [float] A filtered sample

See also:

pysptk.sptk.mglsadf

AllPoleDF

class pysptk.synthesis.**AllPoleDF**(*order=25*)

All-pole digital filter that wraps `poledf`

Attributes

delay [array] Delay

filt(*x*, *coef*)

Filter one sample using using `poledf`

Parameters

x [float] A input sample

coef: array LPC (with loggain)

Returns

y [float] A filtered sample

See also:

pysptk.sptk.poledf

AllPoleLatticeDF

class pysptk.synthesis.**AllPoleLatticeDF**(*order=25*)
All-pole lttice digital filter that wraps `ltcdf`

Attributes

delay [array] Delay

filt(*x*, *coef*)
Filter one sample using using `ltcdf`

Parameters

x [float] A input sample

coef: array PARCOR coefficients (with loggain)

Returns

y [float] A filtered sample

See also:

pysptk.sptk.ltcdf

Synthesis filter interface

class pysptk.synthesis.**SynthesisFilter**
Synthesis filter interface

All synthesis filters must implement this interface

filt(*x*, *coef*)
Filter one sample

Parameters

x [float] A input sample

coef [array] Filter coefficients

Returns

y [float] A filtered sample

3.1.3 Utilities

Audio files

example_audio_file()	Get the path to an included audio example file.

pysptk.util.example_audio_file

pysptk.util.**example_audio_file**()
Get the path to an included audio example file.

Examples

CHAPTER 4

Indices and tables

- genindex
- search

Python Module Index

p

Index

N

P

R

S

T

U