
pysptk Documentation

Release 0.1.0

Ryuichi YAMAMOTO

Aug 03, 2018

Contents

1	Full documentation	3
2	API documentation	5
2.1	API	5
3	Indices and tables	29
	Python Module Index	31

A python wrapper for [Speech Signal Processing Toolkit \(SPTK\)](#).

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)

2.1 API

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

2.1.1 Library routines

<i>agexp</i>	Magnitude squared generalized exponential function
<i>gexp</i>	Generalized exponential function
<i>glog</i>	Generalized logarithmic function
<i>mseq</i>	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

2.1.2 Adaptive cepstrum analysis

<i>acep</i>	Adaptive cepstral analysis
<i>agcep</i>	Adaptive generalized cepstral analysis
<i>amcep</i>	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] Leakage factor. Default is 0.98.
step [float] Step size. Default is 0.1.

tau [float] Momentum constant. Default is 0.9.

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

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

Returns

prederr [float] Prediction error

Raises

ValueError if invalid order of pade approximation is specified

pysptk.sptk.agcep

`pysptk.sptk.agcep()`

Adaptive generalized cepstral analysis

Parameters

x [float] A input sample

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

stage: int -1 / gamma. Default is 1.

lambda_coef [float] Leakage factor. Default is 0.98.

step [float] Step size. Default is 0.1.

tau [float] Momentum constant. Default is 0.9.

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

Returns

prederr [float] Prediction error

Raises

ValueError if invalid number of stage is specified

pysptk.sptk.amcep

`pysptk.sptk.amcep()`

Adaptive mel-cepstral analysis

Parameters

x [float] A input sample

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

alpha: float All-pass constant. Default is 0.35.

lambda_coef [float] Leakage factor. Default is 0.98.

step [float] Step size. Default is 0.1.

tau [float] Momentum constant. Default is 0.9.

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

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

Returns

prederr [float] Prediction error

Raises

ValueError if invalid order of pade approximation is specified

2.1.3 Mel-generalized cepstrum analysis

<i>mcep</i>	Mel-cepstrum analysis
<i>gcep</i>	Generalized-cepstrum analysis
<i>mgcep</i>	Mel-generalized cepstrum analysis
<i>uels</i>	Unbiased estimation of log spectrum
<i>lpc</i>	Linear prediction analysis

pysptk.sptk.mcep

`pysptk.sptk.mcep()`
Mel-cepstrum analysis

Parameters

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

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

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

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

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

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

etype [int]

Type of parameter *eps*

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

Default is 0.

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

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

itype [float]

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

pysptk.sptk.gcep

`pysptk.sptk.gcep()`

Generalized-cepstrum analysis

Parameters

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

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

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

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

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

threshold [float] Threshold in theq. Default is 0.001

etype [int]

Type of parameter *eps*

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

Default is 0.

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

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

itype [float]

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

pysptk.sptk.mgcep

`pysptk.sptk.mgcep()`

Mel-generalized cepstrum analysis

Parameters

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

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

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

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

num_recursions [int] Number of recursions. Default is $\text{len}(\text{windowed}) - 1$.

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

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

threshold [float] Threshold. Default is 0.001.

etype [int]

Type of paramter *e*

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

Default is 0.

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

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

itype [float]

Input data type:

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

Default is 0.

otype [int]

Output data type

0. mel generalized cepstrum: ($c_0 \dots c_m$)
1. MGLSA filter coefficients: $b_0 \dots b_m$
2. $K, c_1 \dots c_m$
3. $K, b_1 \dots b_m$
4. $K, g, c_1 \dots g, c_m$
5. $K, g, b_1 \dots g, b_m$

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

pysptk.sptk.uels

pysptk.sptk.uels()

Unbiased estimation of log spectrum

Parameters

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

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

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

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

threshold [float] Threshold in theq. Default is 0.001

etype [int]

Type of parameter *eps*

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

Default is 0.

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

itype [float]

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

pysptk.sptk.lpc

`pysptk.sptk.lpc()`

Linear prediction analysis

Parameters

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

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

min_det [float] 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

2.1.4 LPC, LSP and PARCOR conversions

<i>lpc2c</i>	LPC to cepstrum
<i>lpc2lsp</i>	LPC to LSP
<i>lpc2par</i>	LPC to PARCOR
<i>par2lpc</i>	PARCOR to LPC
<i>lsp2sp</i>	LSP to spectrum

pysptk.sptk.lpc2c

pysptk.sptk.**lpc2c**()

LPC to cepstrum

Parameters

lpc [array] LPC

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

Returns

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

pysptk.sptk.lpc2lsp

pysptk.sptk.**lpc2lsp**()

LPC to LSP

Parameters

lpc [array] LPC

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

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

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

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

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

otype [int]

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.

pysptk.sptk.lpc2par

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

Parameters

lpc [array] LPC

Returns

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

pysptk.sptk.par2lpc

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

Parameters

par [array] PARCOR

Returns

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

pysptk.sptk.lsp2sp

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

Parameters

lsp [array] LSP

TODO: consider ‘otype’ optional argument

Returns

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

Notes

It is assumed that *lsp* has loggain at *lsp*[0].

Continued on next page

Table 5 – continued from previous page

2.1.5 Mel-generalized cepstrum conversions

<i>mc2b</i>	Mel-cepstrum to MLSA filter coefficients
<i>b2mc</i>	MLSA filter coefficients to mel-cepstrum
<i>c2acr</i>	Cepstrum to autocorrelation
<i>c2ir</i>	Cepstrum to impulse response
<i>ic2ir</i>	Impulse response to cepstrum
<i>c2ndps</i>	Cepstrum to Negative Derivative of Phase Spectrum (NDPS)
<i>ndps2c</i>	Cepstrum to Negative Derivative of Phase Spectrum (NDPS)
<i>gc2gc</i>	Generalized cepstrum transform
<i>gnorm</i>	Gain normalization
<i>ignorm</i>	Inverse gain normalization
<i>frequ</i>	Frequency transform
<i>mgc2mgc</i>	Mel-generalized cepstrum transform
<i>mgc2sp</i>	Mel-generalized cepstrum transform
<i>mgclsp2sp</i>	MGC-LSP to spectrum

pysptk.sptk.mc2b

`pysptk.sptk.mc2b()`

Mel-cepstrum to MLSA filter coefficients

Parameters

mc [array, shape] Mel-cepstrum.

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

Returns

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

pysptk.sptk.b2mc

`pysptk.sptk.b2mc()`

MLSA filter coefficients to mel-cepstrum

Parameters

b [array, shape] MLSA filter coefficients

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

Returns

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

pysptk.sptk.c2acr

`pysptk.sptk.c2acr()`

Cepstrum to autocorrelation

Parameters

c [array] Cepstrum
order [int] Order of cepstrum. Default is $\text{len}(c) - 1$.
fftl [int] FFT length. Default is 256.

Returns

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

Raises

ValueError if non power of 2 *fftl* is specified

pysptk.sptk.c2ir

`pysptk.sptk.c2ir()`
Cepstrum to impulse response

Parameters

c [array] Cepstrum
length [int] Length of impulse response. Default is 256.

Returns

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

pysptk.sptk.ic2ir

`pysptk.sptk.ic2ir()`
Impulse response to cepstrum

Parameters

h [array] Impulse response
order [int] Order of cepstrum. Default is 25.

Returns

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

pysptk.sptk.c2ndps

`pysptk.sptk.c2ndps()`
Cepstrum to Negative Derivative of Phase Spectrum (NDPS)

Parameters

c [array] Cepstrum
fftl [int] FFT length. Default is 256.

Returns

ndps [array, shape (*fftl* // 2 + 1)] NDPS

Raises

ValueError if non power of 2 *fftl* is specified

pysptk.sptk.ndps2c

`pysptk.sptk.ndps2c()`

Cepstrum to Negative Derivative of Phase Spectrum (NDPS)

Parameters

ndps [array, shape (*ffilen* // 2 + 1)] NDPS

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

Returns

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

Raises

ValueError if non power of 2 *ffilen* is detected

pysptk.sptk.gc2gc

`pysptk.sptk.gc2gc()`

Generalized cepstrum transform

Parameters

src_ceps [array] Generalized cepstrum.

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

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

dst_gamma [float] 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

pysptk.sptk.gnorm

`pysptk.sptk.gnorm()`

Gain normalization

Parameters

ceps [array, shape] Generalized cepstrum.

gamma [float] Gamma. Default is 0.0.

Returns

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

Raises

ValueError if invalid *gamma* is specified

pysptk.sptk.ignorm

`pysptk.sptk.ignorm()`

Inverse gain normalization

Parameters

c [array] Normalized generalized cepstrum

gamma [float] Gamma. Default is 0.0.

Returns

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

Raises

ValueError if invalid *gamma* is specified

pysptk.sptk.freqt

`pysptk.sptk.freqt()`

Frequency transform

Parameters

ceps [array] Cepstrum.

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

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

Returns

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

pysptk.sptk.mgc2mgc

`pysptk.sptk.mgc2mgc()`

Mel-generalized cepstrum transform

Parameters

src_ceps [array] Mel-generalized cepstrum.

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

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

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

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

dst_gamma [float] 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

pysptk.sptk.mgc2sp

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

Parameters

- ceps** [array] Mel-generalized cepstrum.
alpha [float] All-pass constant. Default is 0.0.
gamma [float] Gamma. Default is 0.0.
fftlens [int] FFT length. Default is 256.

Returns

sp [array, shape (*fftlens* // 2 + 1)] Complex spectrum

Raises

ValueError

- if invalid *gamma* is specified
- if non power of 2 *fftlens* is specified

pysptk.sptk.mgclsp2sp

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

Parameters

- lsp** [array] MGC-LSP
alpha [float] All-pass constant. Default is 0.0.
gamma [float] Gamma. Default is 0.0.
fftlens [int] FFT length. Default is 256.
gain [bool] Whether the input MGC-LSP should have loggain or not. Default is True.

Returns

sp [array, shape (*fftlens* // 2 + 1)] Complex spectrum

Raises

ValueError

- if invalid *gamma* is specified
- if non power of 2 *fftlens* is specified

2.1.6 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, shape] A whole audio signal

fs [int] Sampling frequency.

hopsize [int] Hop size.

min [float] Minimum fundamental frequency. Default is 50.0

max [float] Maximum fundamental frequency. Default is 800.0

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

otype [int (default=1)] Output format (0) pitch (1) f0 (2) log(f0).

Returns

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

Raises

ValueError if invalid otype is specified

2.1.7 Window functions

<i>blackman</i>	Blackman window
<i>hamming</i>	Hamming window
<i>hanning</i>	Hanning window
<i>bartlett</i>	Bartlett window
<i>trapezoid</i>	Trapezoid window
<i>rectangular</i>	Rectangular window

pysptk.sptk.blackman`pysptk.sptk.blackman()`

Blackman window

Parameters

n [int] Window length

normalize [int]

Normalization flag. 0 : don't normalize 1 : normalize by power 2 : normalize by magnitude

Default is 0.

Returns

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

pysptk.sptk.hamming

`pysptk.sptk.hamming()`

Hamming window

Parameters

n [int] Window length

normalize [int]

Normalization flag. 0 : don't normalize 1 : normalize by power 2 : normalize by magnitude

Default is 0.

Returns

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

pysptk.sptk.hanning

`pysptk.sptk.hanning()`

Hanning window

Parameters

n [int] Window length

normalize [int]

Normalization flag. 0 : don't normalize 1 : normalize by power 2 : normalize by magnitude

Default is 0.

Returns

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

pysptk.sptk.bartlett

`pysptk.sptk.bartlett()`

Bartlett window

Parameters

n [int] Window length

normalize [int]

Normalization flag. 0 : don't normalize 1 : normalize by power 2 : normalize by magnitude

Default is 0.

Returns

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

pysptk.sptk.trapezoid

`pysptk.sptk.trapezoid()`
Trapezoid window

Parameters

n [int] Window length

normalize [int]

Normalization flag. 0 : don't normalize 1 : normalize by power 2 : normalize by magnitude

Default is 0.

Returns

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

pysptk.sptk.rectangular

`pysptk.sptk.rectangular()`
Rectangular window

Parameters

n [int] Window length

normalize [int]

Normalization flag. 0 : don't normalize 1 : normalize by power 2 : normalize by magnitude

Default is 0.

Returns

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

2.1.8 Waveform generation filters

<i>poledf</i>	All-pole digital filter
<i>lmadf</i>	LMA digital filter
<i>lspdf</i>	LSP synthesis digital filter
<i>ltcdf</i>	All-pole lattice digital filter
<i>glsadf</i>	GLSA digital filter
<i>mlsadf</i>	MLSA digital filter
<i>mglسادf</i>	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

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

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

pysptk.sptk.ltcdtf

`pysptk.sptk.ltcdtf()`

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

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

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

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

2.1.9 Utilities for waveform generation filters

<i>poledf_delay</i>	Delay for poledf
<i>lmadf_delay</i>	Delay for lmadf
<i>lspdf_delay</i>	Delay for lspdf
<i>ltcdf_delay</i>	Delay for ltcdf
<i>glsadf_delay</i>	Delay for glsadf
<i>mlsadf_delay</i>	Delay for mlsadf
<i>mglsadf_delay</i>	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

CHAPTER 3

Indices and tables

- `genindex`
- `search`

p

`pysptk, ??`

`pysptk.sptk, 5`

A

acep() (in module pysptk.sptk), 6
agcep() (in module pysptk.sptk), 7
agexp() (in module pysptk.sptk), 5
amcep() (in module pysptk.sptk), 7

B

b2mc() (in module pysptk.sptk), 15
bartlett() (in module pysptk.sptk), 21
blackman() (in module pysptk.sptk), 20

C

c2acr() (in module pysptk.sptk), 15
c2ir() (in module pysptk.sptk), 16
c2ndps() (in module pysptk.sptk), 16

F

frequ() (in module pysptk.sptk), 18

G

gc2gc() (in module pysptk.sptk), 17
gcep() (in module pysptk.sptk), 9
gexp() (in module pysptk.sptk), 6
glog() (in module pysptk.sptk), 6
glsadf() (in module pysptk.sptk), 24
glsadf_delay() (in module pysptk.sptk), 26
gnorm() (in module pysptk.sptk), 17

H

hamming() (in module pysptk.sptk), 21
hanning() (in module pysptk.sptk), 21

I

ic2ir() (in module pysptk.sptk), 16
ignorm() (in module pysptk.sptk), 18

L

lmadf() (in module pysptk.sptk), 23

lmadf_delay() (in module pysptk.sptk), 26
lpc() (in module pysptk.sptk), 12
lpc2c() (in module pysptk.sptk), 13
lpc2lsp() (in module pysptk.sptk), 13
lpc2par() (in module pysptk.sptk), 14
lsp2sp() (in module pysptk.sptk), 14
lspdf() (in module pysptk.sptk), 23
lspdf_delay() (in module pysptk.sptk), 26
ltcdf() (in module pysptk.sptk), 24
ltcdf_delay() (in module pysptk.sptk), 26

M

mc2b() (in module pysptk.sptk), 15
mcep() (in module pysptk.sptk), 8
mgc2mgc() (in module pysptk.sptk), 18
mgc2sp() (in module pysptk.sptk), 19
mgcep() (in module pysptk.sptk), 10
mgclsp2sp() (in module pysptk.sptk), 19
mglsadf() (in module pysptk.sptk), 25
mglsadf_delay() (in module pysptk.sptk), 27
mlsadf() (in module pysptk.sptk), 24
mlsadf_delay() (in module pysptk.sptk), 27
mse() (in module pysptk.sptk), 6

N

ndps2c() (in module pysptk.sptk), 17

P

par2lpc() (in module pysptk.sptk), 14
poledf() (in module pysptk.sptk), 22
poledf_delay() (in module pysptk.sptk), 25
pysptk (module), 1
pysptk.sptk (module), 5

R

rectangular() (in module pysptk.sptk), 22

S

swipe() (in module pysptk.sptk), 20

T

`trapezoid()` (in module `pysptk.sptk`), [22](#)

U

`uels()` (in module `pysptk.sptk`), [11](#)