

Simple Parks-McClellan implementation in Python

Generated by Doxygen 1.9.1

1 Parks-McClellan	1
2 Namespace Index	3
2.1 Namespace List	3
3 Namespace Documentation	5
3.1 parksMcClellan Namespace Reference	5
3.1.1 Detailed Description	6
3.1.2 Function Documentation	7
3.1.2.1 delta()	7
3.1.2.2 filterPlot()	7
3.1.2.3 findExtremal()	7
3.1.2.4 gk()	8
3.1.2.5 Hbp()	8
3.1.2.6 Hexp()	8
3.1.2.7 Hhp()	9
3.1.2.8 Hlp()	9
3.1.2.9 parksMcClellan()	9
3.1.2.10 remez()	11
3.1.2.11 Wconst()	11
3.1.2.12 Wlp()	12
3.1.3 Variable Documentation	12
3.1.3.1 H	12
3.1.3.2 wstop1	12
3.1.3.3 Main	12
Index	13

Chapter 1

Parks-McClellan

Simple implemenation of the Parks-McClellan algorithm for educational purposes

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

parksMcClellan	
Simple implementation of the parksMcClellan algorithm	5

Chapter 3

Namespace Documentation

3.1 parksMcClellan Namespace Reference

Simple implementation of the [parksMcClellan](#) algorithm.

Functions

- def [Wconst](#) (x)
Constant weight function.
- def [Wlp](#) (xpass, xstop, x)
Weight function of a low pass filter.
- def [Hlp](#) (xpass, xstop, x)
Ideal ideal transfer function of a low pass filter.
- def [Hhp](#) (xstop, xpass, x)
Ideal ideal transfer function of a high pass filter.
- def [Hbp](#) (xstop1, xpass1, xpass2, xstop2, x)
Ideal ideal transfer function of a band pass filter.
- def [Hexp](#) (x)
TF of a filter for testing.
- def [gk](#) (k, extremal)
Gamma.
- def [delta](#) (extremal, H, W)
Delta.
- def [findExtremal](#) (E, grid, m, d, debug=True)
Find the extremal points.
- def [remez](#) (F, W, extremal, maxiter=100, eacc=0.0001, wtol=1e-4, debug=True)
remez algorithm
- def [parksMcClellan](#) (H, W, n, maxiter=100, eacc=0.0001, wtol=1e-4, debug=True)
[parksMcClellan](#) algorithm
- def [filterPlot](#) (hk, H, title)
Filter plot.

Variables

- float `wstop1` = $0.3 * \pi$
- float `wpass1` = $0.4 * \pi$
- float `wpass2` = $0.6 * \pi$
- float `wstop2` = $0.7 * \pi$
- int `n` = 50
- def `H`
- `iterations`
- `d`
- `hk`
- `Wconst`
- `debug`
- def `W` = lambda x: `Wlp`(`np.cos(wpass2)`, `np.cos(wstop2)`, x)
- `wtol`

3.1.1 Detailed Description

Simple implementation of the `parksMcClellan` algorithm.

Convergence is painful for in a few conditions. Not quite sure why.

Author

Rodrigo Pedroso Mendes

Version

V1.0

Date

02/05/23 18:31:45

#LICENSE#

Copyright (c) 2023 Rodrigo Pedroso Mendes

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

3.1.2 Function Documentation

3.1.2.1 `delta()`

```
def parksMcClellan.delta (
    extremal,
    H,
    W )
```

Delta.

Parameters

<i>extremal</i>	extremal frequencies
<i>H</i>	ideal transfer function
<i>W</i>	weight function

Returns

real number representing the delta

3.1.2.2 `filterPlot()`

```
def parksMcClellan.filterPlot (
    hk,
    H,
    title )
```

Filter plot.

Parameters

<i>H</i>	ideal transfer function
<i>hk</i>	filter impulse response
<i>title</i>	title

3.1.2.3 `findExtremal()`

```
def parksMcClellan.findExtremal (
    E,
    grid,
    m,
    d,
    debug = True )
```

Find the extremal points.

Parameters

<i>E</i>	pointer to weighted error function
<i>grid</i>	extremal points will be calculated based on E(grid)
<i>m</i>	number of extremal points

Parameters

<i>d</i>	calculated error
<i>debug</i>	enable or disable the debug mode

Returns

array of extremal points

3.1.2.4 gk()

```
def parksMcClellan.gk (
    k,
    extremal )
```

Gamma.

Parameters

<i>k</i>	index
<i>extremal</i>	extremal frequencies

Returns

gamma k

3.1.2.5 Hbp()

```
def parksMcClellan.Hbp (
    xstop1,
    xpass1,
    xpass2,
    xstop2,
    x )
```

Ideal ideal transfer function of a band pass filter.

Parameters

<i>xstop1</i>	1st stop frequency as a function of cos(stop1)
<i>xpass1</i>	1st pass frequency as a function of cos(wpass1)
<i>xpass2</i>	2st pass frequency as a function of cos(wpass2)
<i>xstop2</i>	2st stop frequency as a function of cos(stop2)
<i>x</i>	input vector

Returns

numpy array representing the ideal transfer function

3.1.2.6 Hexp()

```
def parksMcClellan.Hexp (
    x )
```

TF of a filter for testing.

Parameters

<i>x</i>	input vector
----------	--------------

Returns

np array representing the ideal transfer function

3.1.2.7 Hhp()

```
def parksMcClellan.Hhp (
    xstop,
    xpass,
    x )
```

Ideal ideal transfer function of a high pass filter.

Parameters

<i>xstop</i>	stop frequency as a function of cos(stop)
<i>xpass</i>	pass frequency as a function of cos(wpass)
<i>x</i>	input vector

Returns

numpy array representing the ideal transfer function

3.1.2.8 Hlp()

```
def parksMcClellan.Hlp (
    xpass,
    xstop,
    x )
```

Ideal ideal transfer function of a low pass filter.

Parameters

<i>xpass</i>	pass frequency as a function of cos(wpass)
<i>xstop</i>	stop frequency as a function of cos(stop)
<i>x</i>	input vector

Returns

numpy array representing the ideal transfer function

3.1.2.9 parksMcClellan()

```
def parksMcClellan.parksMcClellan (
    H,
    W,
    n,
    maxiter = 100,
    eacc = 0.0001,
    wtol = 1e-4,
    debug = True )
```

[parksMcClellan](#) algorithm

Parameters

<i>H</i>	ideal transfer function
<i>W</i>	weight function
<i>n</i>	order of the filter
<i>maxiter</i>	maximum numbe of iterations
<i>eacc</i>	the algorithm will stop when the error changes between iterations is less than eacc%
<i>wtol</i>	frequency tolerance
<i>debug</i>	enable or disable the debug mode

Returns

error and filter coefficients

3.1.2.10 remez()

```
def parksMcClellan.remez (
    F,
    W,
    extremal,
    maxiter = 100,
    eacc = 0.0001,
    wtol = 1e-4,
    debug = True )
```

remex algorithm

Parameters

<i>F</i>	function to be aproximated
<i>W</i>	weight function
<i>extremal</i>	inital extremal points
<i>maxiter</i>	maximum number of iterations
<i>eacc</i>	the algorithm will stop when the error changes between iterations is less than eacc (%)
<i>wtol</i>	frequency tolerance
<i>debug</i>	enable or disable the debug mode

Returns

error, extremal, lagrange polynomial

3.1.2.11 Wconst()

```
def parksMcClellan.Wconst (
    x )
```

Constant weight function.

Parameters

<i>x</i>	input vector
----------	--------------

Returns

numpy array filled-up with ones

3.1.2.12 Wlp()

```
def parksMcClellan.Wlp (
    xpass,
    xstop,
    x )
```

Weight function of a low pass filter.

Parameters

<i>xpass</i>	pass frequency as a function of cos(wpass)
<i>xstop</i>	stop frequency as a function of cos(stop)
<i>x</i>	input vector

Returns

numpy array representing the weight

3.1.3 Variable Documentation

3.1.3.1 H

```
def parksMcClellan.H
```

Initial value:

```
1 = lambda x: Hbp(np.cos(wstop1), np.cos(wpass1), \
2               np.cos(wpass2), np.cos(wstop2), x)
```

3.1.3.2 wstop1

```
float parksMcClellan.wstop1 = 0.3*np.pi
```

3.1.3.3 Main

Index

- delta
 - [parksMcClellan, 7](#)
- filterPlot
 - [parksMcClellan, 7](#)
- findExtremal
 - [parksMcClellan, 7](#)
- gk
 - [parksMcClellan, 8](#)
- H
 - [parksMcClellan, 12](#)
- Hbp
 - [parksMcClellan, 8](#)
- Hexp
 - [parksMcClellan, 8](#)
- Hhp
 - [parksMcClellan, 9](#)
- Hlp
 - [parksMcClellan, 9](#)
- [parksMcClellan, 5](#)
 - [delta, 7](#)
 - [filterPlot, 7](#)
 - [findExtremal, 7](#)
 - [gk, 8](#)
 - [H, 12](#)
 - [Hbp, 8](#)
 - [Hexp, 8](#)
 - [Hhp, 9](#)
 - [Hlp, 9](#)
 - [parksMcClellan, 9](#)
 - [remez, 11](#)
 - [Wconst, 11](#)
 - [Wlp, 11](#)
 - [wstop1, 12](#)
- remez
 - [parksMcClellan, 11](#)
- Wconst
 - [parksMcClellan, 11](#)
- Wlp
 - [parksMcClellan, 11](#)
- wstop1
 - [parksMcClellan, 12](#)