Simple Parks-McClellan implementation in Python

Generated by Doxygen 1.9.1

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:

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)

  remex 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∗np.pi
- float **wpass1** = 0.4∗np.pi
- float **wpass2** = 0.6∗np.pi
- float **wstop2** = 0.7∗np.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#

3.1.2 Function Documentation

3.1.2.1 delta()

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

Delta.

Parameters

extremal	extremal frequencies
H	ideal transfer function
W	weight function

Returns

real number representing the delta

3.1.2.2 filterPlot()

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

Filter plot.

Parameters

H	ideal transfer function
hk	filter impulse response
title	title

3.1.2.3 findExtremal()

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

Find the extremal points.

Parameters

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

Parameters

d	calculated error
debug	enable or disable the debug mode

Returns

array of extremal points

3.1.2.4 gk()

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

Gamma.

Parameters

k	index
extremal	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

xstop1	1st stop frequency as a function of cos(stop1)
xpass1	1st pass frequency as a function of cos(wpass1)
xpass2	2st pass frequency as a function of cos(wpass2)
xstop2	2st stop frequency as a function of cos(stop2)
x	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

x	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

xstop	stop frequency as a function of cos(stop)
xpass	pass frequency as a function of cos(wpass)
x	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

xpass	pass frequency as a function of cos(wpass)
xstop	stop frequency as a function of cos(stop)
x	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

H	ideal transfer function
W	weight function
n	order of the filter
maxiter	maximum numbe of iterations
eacc	the algorithm will stop when the error changes between iterations is less than eacc%
wtol	frequency tolerance
debug	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

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

Returns

error, extremal, lagrange polynomial

3.1.2.11 Wconst()

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

Constant weight function.

Parameters

x	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

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

Returns

numpy array representing the weight

3.1.3 Variable Documentation

3.1.3.1 H

```
def parksMcClellan.H
```

Initial value:

```
=  lambda x: Hbp(np.cos(wstop1), np.cos(wpass1), \
                     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