

DRFilter 滤波器用法

DRFilter 滤波器控件提供一个信号带通滤波器模块，可以对收到的采样信号进行带通滤波。该控件为后台控件，在 GUI 界面上无显示。

在 Python 程序 Tkinter GUI 界面中创建一个 DRFilter 滤波器控件的命令和语法格式如下：

```
hl=DRFilter(option=value, ...)
```

该控件创建时只 10 个可选的输入参数：

mt: 信号滤波算法，默认值=0, 0/1/2=FFT/Fir/IIR

Fs: 信号采样频率，默认值=44100，实际值=滤波信号采样频率

f1: 带通滤波器下限频率，默认值=0，

f2: 带通滤波器上限频率，默认值=8000Hz，，消高于 8000Hz 的高频噪声

firM: FIR 滤波器系数长度，默认值 201

firWin: FIR 滤波器设计采用的窗函数，默认值'hamming'。参见 Scipy 库 Window types。

iirMT: IIR 滤波器类型，默认值 0; 0/1/2/3=Butter/cheb1/cheb2/ellip

iirP: IIR 滤波器阶次，默认值 8

Irrp: 通带邹波幅度，默认值 1db，仅 cheb1 和 ellip 滤波器用到

Iirrs:阻带邹波衰减幅度，默认值，40db，仅 cheb2 和 ellip 滤波器用到

返回值 hl 则是该控件创建后的句柄变量名，可以通过 hl 对创建后的控件进行进一步的设定。

DRFilter 滤波器控件有 6 个属性函数，其功能定义如下：

hl.setValue2DX(dx,yarray): 设定滤波信号采样间隔和采样数组

hl.setType(mt): 设定滤波器类型

hl.setBand(f1,f2): 设定滤波器频率通带

hl.setFIR(firM,firWin): 设定采用 FIR 滤波器时的参数

hl.setIIR(iirMT,iirP,iirrp,iirrs): 设定采用 IIR 滤波器时的参数

dt,yarrayFiltered=hl.getValue2DX(): 获取滤波后的信号

下面是一个 DRFilter 滤波器控件创建实例：

```
hl=dsp.DRFilter()
```

通常，可以选择设定部分选择参数的形式进行设定，其余可选参数采用默认值。创建实例如下：

```
hl=dsp.DRFilter(f1=0,f2=500)
```

创建后可以通过返回的控件句柄对其属性函数进行操作。

DRFilter 滤波器控件可以通过绑定的方式将滤波后的信号数据传递给绑定的回调函数或其他控件。获取数据的绑定方法为：

```
hl.addCallBack2DX(fun)
```

被绑定的回调函数或另一控件的 set 函数是一个单值+一维数组函数，收到的是信号曲线的采样间隔 dt 和 y 数组值。回调函数形式为：

```
def setArrayTwoX(dt,yArray):
```

```
...
```

当采用 DRPython 进行图形化编程时，控件的创建代码和直连绑定代码的生成均可以由机器实现；控件创建只需要将所需控件拖拽到 GUI 布局区，而直连绑定则需要在 GUI 布局区的控件间进行连线操作。DRFilter 滤波器控件可以做为字符串或单数值输出控件，在连线状态下在控件上点击鼠标，则弹出端选择；选择完成后，点击另一控件，则弹出输入端选择；若输出端和输入端类型一致，则完成连线。输出参数为单值+一维数组构成的滤波信号数据(addCallBack2DX)。



DRFilter 滤波器控件也可以做为单数/字符串输入控件，在连线状态下当上一控件已选输出函数后，在控件上点击鼠标，则弹出连线选择菜单；选择完成后，若输出端和输入端类型一致，则完成连线。输入参数为单值+一维数组的输入信号(setValue2DX)。



下面是 DRFilter 滤波器控件的创建和应用示例小程序，功能是在 GUI 界面上插入 1 个按钮控件，1 个信号发生器控件、1 个 DRFilter 滤波器控件，2 个 Plot 控件，然后分别通过直连绑定的方式，用按钮控件驱动信号发生器产生扫频波信号，驱动滤波器控件进行带通滤波，然后进行显示。DRFilter 滤波器控件应用示例小程序代码如下：

```
#==引用的包
import tkinter as tk
import numpy as np
import drvi.drviControlls as dr
import drvi.drviDSP as dsp

#用户自定义的回调函数
def muserSingle(v):
    global f1
    Fs=5120;
    N=1024;
    t=np.arange(N)*(1.0/Fs)
    x=np.sin(2*np.pi*(100+2000*t)*t)
    mPlot1.setValue2D(t,x)
    abc.setType(2)
    f2=f1+200
    abc.setBand(f1,f2)
    abc.setValue2DX(1/Fs,x)
    dt,x1=abc.getValue2DX()
```

```

        mPlot2.setValue2DX(dt, x1)
        f1=f1+100
        if f1>800: f1=800

#==创建 Tkinter GUI 窗口
win= tk.Tk()
win.geometry('1100x600')
win.config(bg="#ddeeee")
#==插入控件
mButton1=dr.DRButton(win, 20, 10, 100, 30, txt=' Run' )
abc=dsp.DRFilter()
mPlot1=dr.DRPlot(win, 20, 60, 800, 210)
mPlot2=dr.DRPlot(win, 20, 300, 800, 210)
#==信号的直连绑定
mButton1.addCallBackSingle(muserSingle)
#==Global
f1=0
#==主程序循环
win.mainloop()

```

pyqt 版本中的用法说明：

DRVI 有继承自 Tkinter 和继承自 pyqt 两个不同库，除引用的 GUI 库(Tkinter/pyqt)和框架程序略有不同外，DRVI 部分控件的创建、布局和连线是完全相同的。因均继承自 Numpy 和 Scipy，该函数在两个版本中用法无区别。