

## DRSpectgram 谱阵分析用法

DRSpectgram 谱阵分析控件提供一个 STFT 信号分析模块，可以对收到的采样信号进行短时傅里叶变换谱阵分析。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 DRSpectgram 谱阵分析控件的命令和语法格式如下：

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

该控件创建时只有一组可选的输入参数，包括：

**nfft**：每帧数据长度和 FFT 变换长度，默认值=1024

**overlap**：帧间重叠长度，默认值=128，即  $nfft/8$

**mt**：返回谱阵数据类型，默认值=0；0/1/2=功率谱/幅值谱/对数功率谱

**mclip**：对数功率谱时的数据动态范围，默认值：140db，过大谱图显示效果不好

**Pre**：信号预加重开关，默认值=0，0/1=关/开；预加重即信号谱阵分析前加了一个差分处理(高通滤波)，以加重信号中的高频成分。

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。

DRSpectgram 谱阵分析控件有 4 个属性函数，其功能定义如下：

**hl.setValue2DX(dx,yarray)**：设定谱阵分析信号的采样间隔和采样数组

**tarray,farray,data2D=hl.getArray3D()**：获取当前谱阵计算数据

**hl.setPar(nfft,overlap)**：设定 STFT 计算参数

**hl.setVSType(mt)**：设定谱阵计算类型，0/1/2=功率谱/幅值谱/对数功率谱

**hl.setVSPre(pre)**：设定预加重开关，0/1=关/开

下面是一个全参数的 DRSpectgram 谱阵分析控件创建实例：

```
hl=dsp.DRSpectgram()
```

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.DRSpectgram(mt=2)
```

创建后可以通过返回的控件句柄对其属性函数进行操作。

DRSpectgram 谱阵分析控件可以通过绑定的方式信号的频谱分析数据传递给绑定的回调函数或其他控件。获取幅值谱的计算数据绑定方法为：

```
hl.addCallBack3D(fun)
```

被绑定的回调函数或另一控件的 set 函数是一个由 3 个数组参数组成的函数，收到分别是谱阵的时间点，频率点和谱阵数据。回调函数形式为：

```
def setValue3D(tArray,fArray,stft2DArray):
```

...

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。DRSpectgram 谱阵分析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 2 个一维数组和一个 2 维数组构成的谱阵数据(addCallBack3D)。



DRSpectgram 谱阵分析控件也可以做为单值/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+一维数组的采样信号(setValue2DX)，单值的谱阵类型(setVSType)，单值的数据预加重开关(setVSPre)。



下面是 DRSpectgram 谱阵分析控件的创建和应用示例小程序，功能是在 GUI 界面上插入一个按钮控件、一个 DRSpectgram 谱阵分析控件、一个 DRColormesh 语谱图显示控件和一个用户自定义函数，然后通过直连绑定方式，用按钮控件来驱动用户自定义回调函数产生一段 chirp 信号，然后用 DRSpectgram 计算谱阵，最后用 DRColormesh 语谱图显示控件来显示。DRSpectgram 谱阵分析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
from scipy import signal
import drvi.drviControlls as dr
import drvi.drviDSP as dsp

#用户自定义的回调函数
def muserSingle3(v):
    Fs=5120;
    N=50000;
    t=np.arange(N)*(1.0/Fs)
    y=np.sin(2*np.pi*(100+5*t*t)*t)
    abc.setValue2DX(1/Fs,y)
    tt,ff,Sxx=abc.getArray3D()
    mArray.setValue3D(tt,ff,Sxx)

#==创建 Tkinter GUI 窗口
win= tk.Tk()
win.geometry('900x497')
win.config(bg="#ddeeef")
#==插入按钮控件
```

```
mButton1=dr.DRButton(win,20,20,100,30,txt='Run')
#==插入语谱图显示控件
mArray=dr.DRColormesh(win,20,60,800,400)
#==插入谱阵计算控件
abc=dsp.DRSpectgram(mt=2)
#==绑定用户回调函数
mButton1.addCallbackSingle(muserSingle3)
#==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。