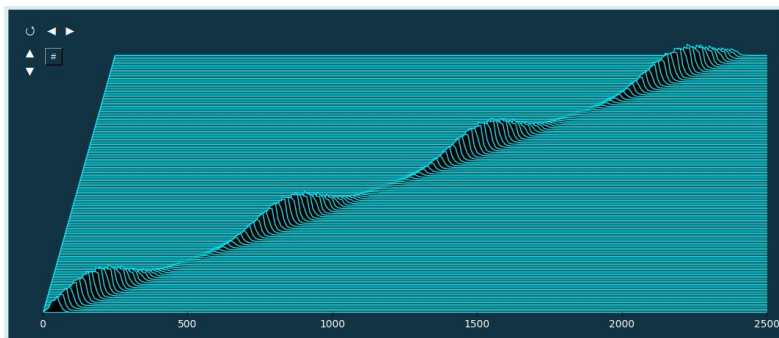


DRwaterFall 谱阵显示控件用法

DRwaterFall 谱阵显示控件主要用于显示时变信号的谱阵图。一般来说，幅值谱阵的显示效果较好；另外，谱阵的阶数通常控制在 100 阶以下，过高显示效果也不好。在阵图显示方面，其作用与 DRPlotSurface 三维曲面控件相似，但因采用二维图象模式，显示效率更高；另外 DRPlotSurface 所依托的 matplotlib 库中 plot_surface 函数对谱阵形成的伪三维曲面的支持不好，显示效果欠佳。



DRwaterFall 谱阵显示控件图形显示效果图

在 Python 程序 Tkinter GUI 界面中创建一个 DRwaterFall 谱阵显示控件的命令和语法格式如下：

```
hl= DRwaterFall(win,x,y,w,h,option=value, ...)
```

其中输入参数 win 为 Tkinter 库创建的 GUI 窗口句柄，[x,y,w,h]四个参数为该控件插入 GUI 窗口 win 时的左上角坐标[x,y]和该控件的宽、高[w,h]。另外是一组可选的输入参数，包括：

cb1: 背景色，默认值='#113344'

cb2: 背景色 1，默认值='#000000'

cf1: 前景色，默认值='#ffffff'

cf2: 前景色，默认值='#00eeff'

xDist: 阵图横移系数，默认值=0.1

yDist: 阵图顶边距系数，默认值=0.1

zDist: 阵图幅值比例系数，默认值=0.1

flow: 横坐标最小值，默认值=0

fhigh: 横坐标最大值，默认值=0

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。控件创建时，当 flow 和 fhigh 均设为零时，横坐标轴采用自动量程。

DRwaterFall 谱阵显示控件有 2 个属性函数，其功能定义如下：

hl.setValue3D(t,f,a2D): 设定谱阵曲线数据，t 和 f 为一维数组，a2D 为二维数组，对应
f,t,a2D=signal.spectrogram(...)

`hl.addFrame(M,f,A,d)`: 添加一帧谱阵曲线, `M` 为谱阵曲线的总阶数, `f` 推入频谱的频率点数据, `A` 为频谱数据, 对应: `f,A=dsp.AmplitudeSpectrum(...)`; `d` 为推入方向, 取 0/1。

下面是一个全参数的 `DRwaterFall` 谱阵显示控件创建实例:

```
hl=dr. DRwaterFall(win,54,104,1000,400,'#113344','#000000','#ffffff','#00eeff',0.1,0.1,0.1,0,0)
```

通常, 可以选择设定部分选择参数的形式进行设定, 除基本参数 `win`、`[x,y,w,h]`和用变量名设定的可选参数外, 其余可选参数采用默认值。创建实例如下:

```
hl=dr. DRwaterFall(win,54,104,1000,400,xDist=0.05)
```

创建后可以通过返回的控件句柄对其属性函数进行操作。

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。`DRwaterFall` 谱阵显示控件为输入控件, 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。输入端口(`setValue3D`)收到的是控件的曲线显示数据。



下面是 `DRwaterFall` 谱阵显示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 `DRwaterFall` 谱阵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, 然后计算 `STFT` 谱阵, 最后用 `DRwaterFall` 谱阵显示控件来显示。`DRwaterFall` 谱阵显示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

#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)
    nft=1024; ovl=int(nft/2)
    ff, tt, Sxx=signal.spectrogram(y, Fs, nperseg=nft, noverlap=ovl)
```

```

mArray.setValue3D(tt, ff, Sxx)

#==创建 Tkinter GUI 窗口
win= tk.Tk()
win.geometry('900x497')
win.config(bg="#ddeeee")
#==插入按钮控件
mButton1=dr.DRButton(win, 20, 20, 100, 30, txt=' Run' )
#==插入谱阵显示控件
mArray=dr. DRwaterFall (win, 20, 60, 800, 400)
#==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 Matplotlib，该函数在两个版本中用法无区别。