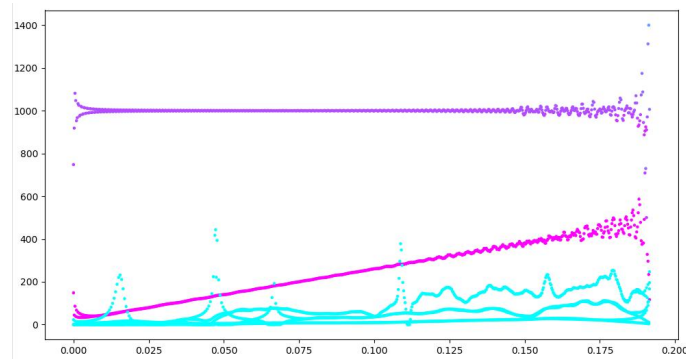


DRScatter 散点图控件用法

DRScatter 散点图控件主要用于显示二维数组定义的散点图；也可以用于显示三维数组定义的散点图，此时 z 坐标对应的幅值变化映射为不同的色彩。



DRScatter 散点图控件图形显示效果图

在 Python 程序 Tkinter GUI 界面中创建一个 DRScatter 散点图控件的命令和语法格式如下：

```
hl=DRScatter(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='#ffffff'

cf1: 前景色，默认值='#000000'，座标色

cf2: 前景色，默认值='#00eeff'，曲线色

cmap: 色彩映射集，默认值='cool'，参略 matplotlib 库 cmap 参数

dot: 散点直径，默认值=3；

xlow: 横坐标最小值，默认值=0

xhigh: 横坐标最大值，默认值=0

ylow: 纵坐标最小值，默认值=0

yhigh: 纵坐标最大值，默认值=0

zlow: 色彩坐标最小值，默认值=0

zhhigh: 色彩坐标最大值，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，当相应轴 low 和 xhigh 均设为零时，采用自动量程。

DRScatter 散点图控件有 4 个属性函数，其功能定义如下：

hl.setValue2D(xarray,yarray,c=None): 设定二维散点数据，并以覆盖形式重新绘制，c 为 16 进制色彩码，如'#ff0000'

hl.addValue2D(xarray,yarray,c=None): 设定二维散点数据，并以添加形式绘制，c 为 16 进制色彩码，如'#ff0000'

hl.setValue3DC(xarray,yarray,carray): 用三个数组设定散点图数据，第三个数组映射为色彩变换，以覆盖形式重新绘制。

hl.addValue3DC(xarray,yarray,carray): 用三个数组设定散点图数据，第三个数组映射为色彩变换，以添加形式绘制。

下面是一个全参数的 DRScatter 散点图控件创建实例：

```
hl=dr.DRScatter(win,143,119,700,300,'#ffffff','#000000','#00eeff','cool',5,0,0,0,0,0,0)
```

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.DRScatter(win,143,119,700,300,dot=3)
```

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。DRScatter 散点图控件为输入控件，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(setValue2D)收到的是控件的曲线显示数据。



下面是 DRScatter 散点图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，一个用户自定义函数和一个 DRScatter 散点图控件，然后通过直连绑定方式，用按钮控件来控制自定义函数生成仿真数据，然后用 DRScatter 散点图控件来显示，同时将信号幅值本身作为色彩，从而绘制出渐变色的曲线。DRScatter 散点图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

def userFun(v):
    global ff;
    Fs=1000
    N=1000
    ff=ff+2
    t=np.arange(N)*(1/Fs)
    y=np.sin(2*np.pi*ff*t)
    mP3.setValue3DC(t,y,y)

##创建 Tkinter GUI 窗口
win= tk.Tk()
win.geometry('900x500')
win.config(bg="#ddeeee")
##插入按钮控件
mB1=dr.DRButton(win,78,39,100,40,'#00aacc','#eeee00','Button',12,101)
##插入 Scatter 控件
mP3=dr.DRScatter(win,95,122,700,210)
##控件直连绑定
mB1.addCallBackSingle(userFun)
##global var===
ff=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 Matplotlib，该函数在两个版本中用法无区别。