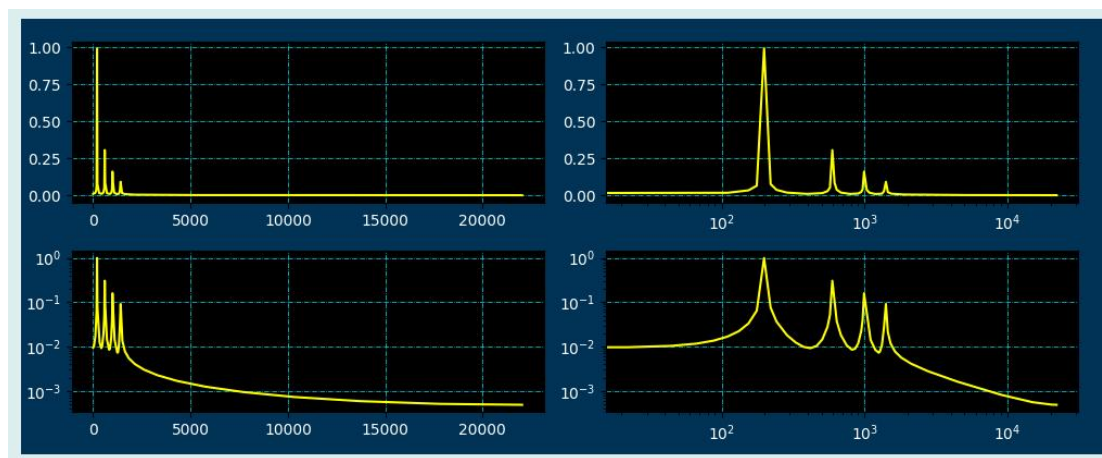


DRSubPlot 曲线控件用法

DRSubPlot 曲线控件主要用于显示多条信号波形、频谱等二维曲线。因做了显示优化，与几个独立的 Plot 控件相比，在显示不断动态刷新的多条曲线时，效果要好一些；但因曲线安排紧凑，无法安放幅值调整等控件。



DRSubPlot 曲线控件图形显示效果图

在 Python 程序 Tkinter GUI 界面中创建一个 DRSubPlot 曲线控件的命令和语法格式如下：

`hl=DRSubPlot(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='#003355'

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'

cf3: 前景色，默认值='#ffff00'

N: 曲线图的行数，默认值=2

M: 曲线图的列数，默认值=2

Style: 各曲线图的曲线格式，默认值=[0,0,0,0]；其中 0=plot, 1=stem, 2=semix, 3=semy, 4=loglog
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。

DRSubPlot 曲线控件有 7 个属性函数，其功能定义如下：

`hl.setXlim(k,xlow,xhigh)`: 设定子曲线图 k 的 X 坐标显示范围，k 为图号，取值 1 到 8

`hl.setYlim(k,ylow,yhigh)`: 设定子曲线图 k 的 Y 坐标显示范围，k 为图号，取值 1 到 8

`hl.setValue2DX(k,xarray,yarray)`: 设定子曲线图 k 的数据，并绘制曲线图。k 为图号，取值 1 到 8

`hl.setValue2D1(xarray,yarray)`: 设定子曲线图 1 的数据；

`hl.setValue2D2(xarray,yarray)`: 设定子曲线图 2 的数据；

`hl.setValue2D3(xarray,yarray)`: 设定子曲线图 3 的数据；

`hl.setValue2D4(xarray,yarray)`: 设定子曲线图 4 的数据；

下面是一个全参数的 DRSubPlot 曲线控件创建实例：

```
hl=dr.DRSubPlot(win,61,112,700,400,'#003355','#000000','#fffff','#00eeff','#ffff00',2,1,[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.DRSubPlot(win,62,80,1000,400,N=2,M=2,style=[0,2,3,4])
```

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。DRSubPlot 曲线控件为输入控件。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。4 个输入项(setValue2D1、setValue2D2、setValue2D3、setValue2D4)分别对应于控件中显示的 4 条曲线。



下面是 DRSubPlot 曲线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、一个 DRSubPlot 曲线控件，然后通过直连绑定方式，用按钮控件来调用用户自定义函数，生成测试信号和信号的幅值谱，然后在 subplot 上用 plot、semix、semy 和 loglog 四种不同形式来显示信号的幅值谱。DRSubPlot 曲线控件应用示例小程序代码如下：

```
##引用的包
##引用的包
import tkinter as tk
import drvi.drviControls as dr
import numpy as np
from scipy.fftpack import fft,ifft

#用户自定义的回调函数
def muserSingle3(v):
    ##生成多频测试信号
    Fs=44100;
    N=2000;
    dt=1.0/Fs
    t=np.arange(N)*dt
    x1=np.sin(2*np.pi*200*t)
    x1=x1+0.33*np.sin(2*np.pi*600*t)
    x1=x1+0.2*np.sin(2*np.pi*1000*t)
    x1=x1+0.14*np.sin(2*np.pi*1400*t)
    ##计算信号的幅值谱
    y2=np.abs(fft(x1))
```

```

f=np.arange(int(N/2))*Fs/N;
a2=2*y2[:int(N/2)]/N
#==用 Subplot 显示幅值谱
mSub.setValue2DX(1,f,a2)
mSub.setValue2DX(2,f,a2)
mSub.setValue2DX(3,f,a2)
mSub.setValue2DX(4,f,a2)

#==创建 Tkinter GUI 窗口
win= tk.Tk()
win.geometry('1000x497')
win.config(bg="#ddeeee")
#==插入开关按钮控件
mButton1=dr.DRSwitch(win,50,22,80,40)
#==插入 Subplot, 2x2, plot, semix, semiy, loglog
mSub=dr.DRSubPlot(win,62,80,1000,400,N=2,M=2,style=[0,2,3,4])
#==直连方式绑定
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。