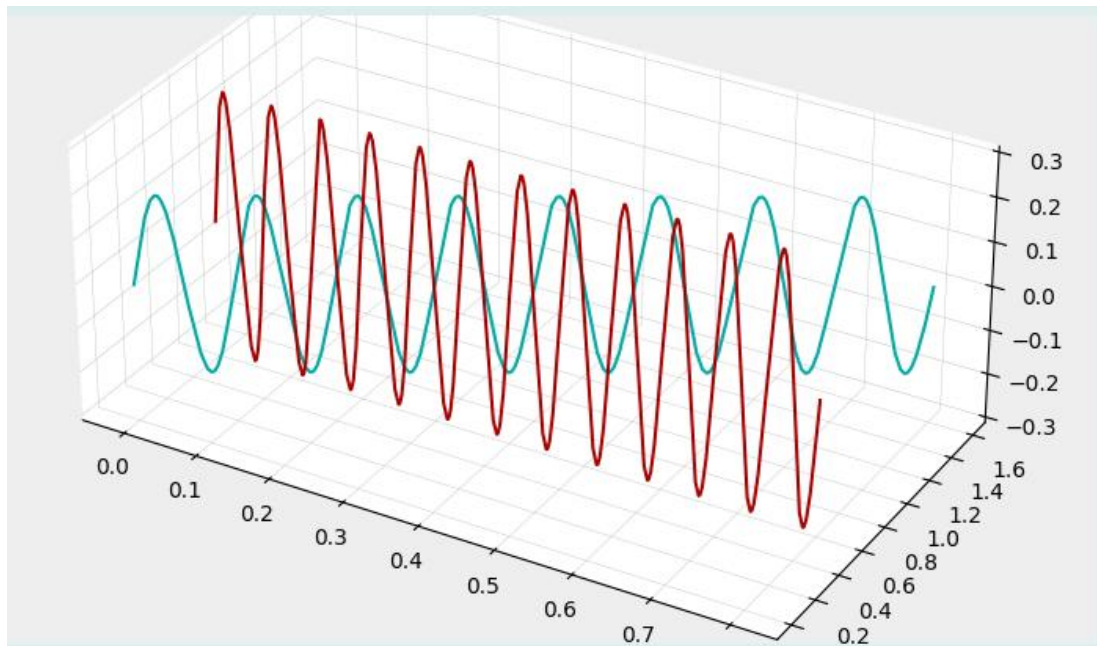


DRPlot3D 三维曲线控件用法

DRPlot3D 三维曲线控件主要用于在三维空间内显示信号曲线。



DRPlot3D 三维曲线控件图形显示效果图

在 Python 程序 Tkinter GUI 界面中创建一个 DRPlot3D 三维曲线控件的命令和语法格式如下：

```
hl=DRPlot3D(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='#ddeeee'

cb2: 坐标平面背景色，默认值='#ffffff'

cf1: 文字色，默认值='#000000'

cf2: 网格线色，默认值='#bbbbbb'

xarray: 曲线 x 坐标数据数组，默认值=0

yarray: 曲线 y 坐标数据数组，默认值=0

zarray: 曲线 z 坐标数据数组，默认值=0

xlow: X 轴最小值，默认值=0;

xhigh: X 轴最大值，默认值=0; 若 xlow=xhigh, 用自动量程

ylow: Y 轴最小值，默认值=0;

yhigh: Y 轴最大值，默认值=0; 若 ylow=yhigh, 用自动量程

zlow: Z 轴最小值，默认值=0;

zhigh: Z 轴最大值，默认值=0; 若 zlow=zhigh, 用自动量程

rate: x,y,z 轴绘图比例调整，默认值=1, 0/1=不调整/自动调整

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。

DRPlot3D 三维曲线控件有 2 个属性函数，其功能定义如下：

hl.setValue3D(xarray,yarray,zarray,c='#009999',s=0): 设定三维曲线数据，并以覆盖形式重新绘制饼图，s=0/1 对应 plot/stem。

hl.addValue3D(xarray,yarray,zarray,c='#aa0000',s=0): 设定三维曲线数据，并以添加形式绘制新曲线，s=0/1 对应 plot/stem

下面是一个全参数的 DRPlot3D 三维曲线控件创建实例：

```
hl=dr.DRPlot3D(win,213,40,420,380,"",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.DRPlot3D(win,213,40,420,380)
```

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。DRPlot3D 三维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。输入端口(setValue3D)收到的是控件的曲线显示数据。



下面是 DRPlot3D 三维曲线控件的创建和应用示例小程序，功能是在 GUI 界面上插入一个用户自定义函数、一个 DRPlot3D 三维曲线控件，然后调用用户自定义函数生成数据和显示。DRPlot3D 三维曲线控件应用示例小程序代码如下：

```
#==引用的包
import tkinter as tk
import drvi.drviControllls as dr
import numpy as np

#用户自定义的回调函数=====
def muserSingle3(v):
    Fs=5120
    dt=1.0/Fs
    N=4096
    t=np.arange(N)*dt
    z1=0.2*np.sin(2*np.pi*10*t)
    z2=0.3*np.sin(2*np.pi*15*t)
    y=np.ones(N)*0.6
    mP3D.setValue3D(t,2*t,z1,'#00aaaa')
    mP3D.addValue3D(t,y,z2,'#aa0000')

#==创建 Tkinter GUI 窗口
win= tk.Tk()
win.geometry('816x497')
win.config(bg="#ddeeee")
#==插入控件
```

```
mP3D=dr.DRPlot3D(win, 20, 40, 700, 400)
#==直连方式绑定
muserSingle3(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。