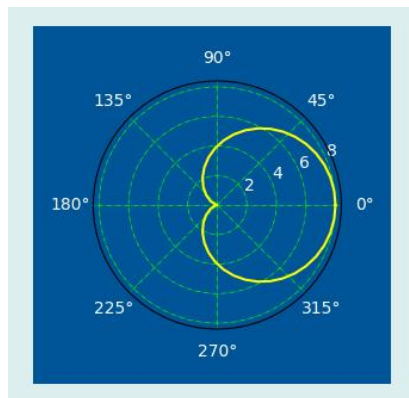


DRPlotPolar 极坐标图控件用法

DRPlotPolar 极坐标图控件主要用于显示极坐标形式的二维曲线。



DRPlotPolar 极坐标图控件图形显示效果图

在 Python 程序 Tkinter GUI 界面中创建一个 DRPlotPolar 极坐标图控件的命令和语法格式如下：

```
hl=DRPlotPolar(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='#005599'

cf1: 前景色，默认值='#ffffff'

cf2: 前景色，默认值='#00ff00'

cf3: 前景色，默认值='#ffff00'

title: 曲线标题，默认值=""

tarray: 曲线角度坐标数据数组，默认值=0

rarray: 曲线极径坐标数据数组，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。

DRPlotPolar 极坐标图控件有 2 个属性函数，其功能定义如下：

hl.setValue2D(tarray,rarray): 设定极坐标图数据，并以覆盖形式重新绘制饼图

hl.addValue2D(tarray,rarray): 设定极坐标图数据，并以添加形式绘制新曲线。

下面是一个全参数的 DRPlotPolar 极坐标图控件创建实例：

```
hl=dr.DRPlotPolar(win,225,124,300,300,'#005599','#ffffff','#00ff00','#ffff00',"",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.DRPlotPolar(win,70,105,700,210)
```

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。DRPlotPie 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 DRPlotPolar 极坐标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 DRPlotPolar 极坐标图控件，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 DRPlotPolar 极坐标图控件来显示。DRPlotPolar 极坐标图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):
    r = np.arange(0, 2, 0.01)
    theta = 2 * np.pi * r
    mPolar.setValue2D(theta, r)

##创建 Tkinter GUI 窗口
win= tk.Tk()
win.geometry('816x497')
win.config(bg="#ddeeee")
##插入开关按钮控件
mButton1=dr.DRSwitch(win, 50, 42, 80, 40)
##插入极坐标图控件
mPolar=dr.DRPlotPolar(win, 225, 124, 300, 300)
##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。