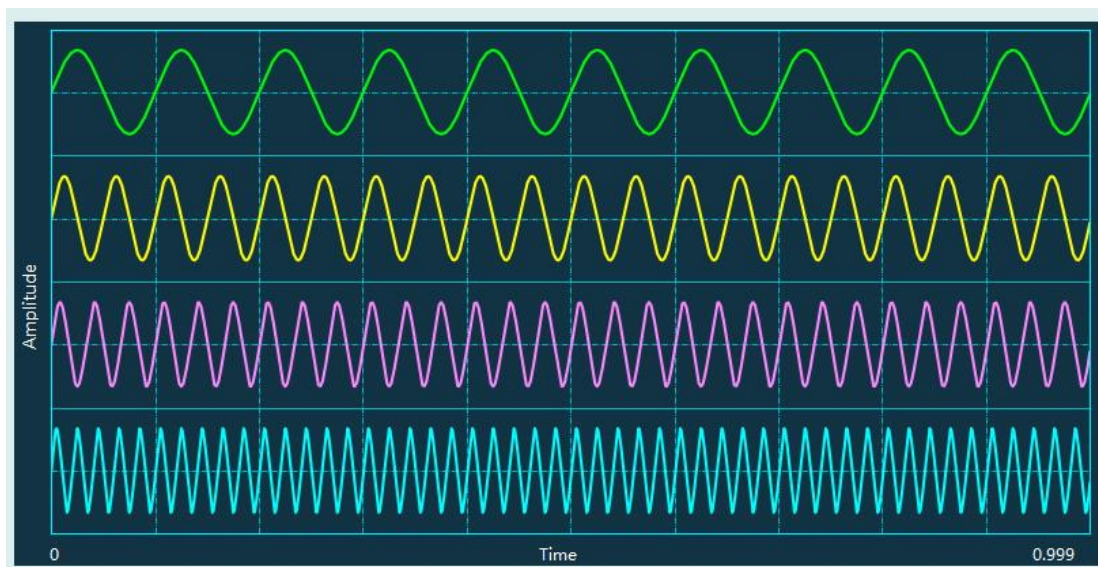


DRMonitor 多曲线显示控件主要用于显示多条信号波形曲线和信号的小波多尺度分解曲线等。因曲线显示在同一个示波控件上，画面利用率更高，约束是曲线的类型必须一致。



DRMonitor 多曲线显示控件图形显示效果图

在 Python 程序 Tkinter GUI 界面中创建一个 DRMonitor 多曲线显示控件的命令和语法格式如下：

```
hl=DRMonitor(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'

cf1: 前景色，默认值='#ffffff'

cf2: 前景色，默认值='#00eeff'

xlabel: X 坐标轴标题，默认值='Time'

ylabel: Y 坐标轴标题，默认值='Amplitude'

T: X 坐标轴显示窗口宽度，默认值=0，0=取信号窗口宽度

wn: 曲线区数量或曲线条数，默认值=4

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。

DRMonitor 多曲线显示控件有 7 个属性函数，其功能定义如下：

hl.setValue2DM(M,tarray,yarray,c='#ffff00',x0=0,ylow=0,yhigh=0,cl=0): 设定子曲线图 M 的曲线数据和绘图参数。cl=0/1，代表覆盖和添加曲线绘制模式，默认为覆盖模式。

hl.setValue2DXM(M,dt,yarray,c='#ffff00',x0=0,ylow=0,yhigh=0,cl=0): 设定子曲线图 M 的曲线数据和绘图参数。

hl.setValue2D1(xarray,yarray): 设定子曲线图 1 的数据，用于快速直连；

hl.setValue2D2(xarray,yarray): 设定子曲线图 2 的数据，用于快速直连；

hl.setValue2D3(xarray,yarray): 设定子曲线图 3 的数据，用于快速直连；

hl.setValue2D4(xarray,yarray): 设定子曲线图 4 的数据, 用于快速直连;

hl.setValue2DX1(dt,yarray): 设定子曲线图 1 的数据, 用于快速直连;

hl.setValue2DX2(dt,yarray): 设定子曲线图 2 的数据, 用于快速直连;

hl.setValue2DX3(dt,yarray): 设定子曲线图 3 的数据, 用于快速直连;

hl.setValue2DX4(dt,yarray): 设定子曲线图 4 的数据, 用于快速直连;

hl.setXWidth(T): 用于更改 X 坐标的显示窗口宽度

hl.clearFig(wn=None): 清空整个曲线绘图区。通常当使用添加模式绘制曲线后, 下一次开始重新绘制整个图形时, 可使用该函数清空画面; 若未使用添加模式, 则新曲线会自动覆盖旧曲线, 此时不用该函数。

wn 为曲线区数量或曲线条数, 默认为继承控件创建时的曲线数。

下面是一个全参数的 DRMonitor 多曲线显示控件创建实例:

```
hl=dr.DRMonitor(win,136,62,700,300,'#113344','#ffffff','#00eeff','Time','Amplitude',0,4)
```

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.DRMonitor(win,136,62,700,300,wn=2)
```

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。DRMonitor 多曲线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。4 个输入项(setValue2D1、setValue2D2、setValue2D3、setValue2D4)和另 4 个输入项(setValue2DX1、setValue2DX2、setValue2DX3、setValue2DX4)分别对应于控件中显示的 4 条曲线。



下面是 DRMonitor 多曲线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, 一个用户自定义函数、一个 DRMonitor 多曲线显示控件, 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，然后在 DRMonitor 显示。DRMonitor 多曲线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
import drvi.drviControllls as dr

#用户自定义的回调函数
def muserSingle(v):
    global ff;
    Fs=1000
    N=1000
    ff=ff+10
    t=np.arange(N)*(1/Fs)
    y1=np.sin(2*np.pi*ff*t)
    y2=np.sin(2*np.pi*2*ff*t)
    y3=np.sin(2*np.pi*3*ff*t)
    y4=np.sin(2*np.pi*5*ff*t)
    mPlot.setValue2DM(0, t, y1, '#00ff00', 0, -1.5, 1.5)
    mPlot.setValue2DM(1, t, y2, '#ffff00', 0, -1.5, 1.5)
    mPlot.setValue2DM(2, t, y3, '#ff88ff', 0, -1.5, 1.5)
    mPlot.setValue2DM(3, t, y4, '#00ffff', 0, -1.5, 1.5)

#==创建 Tkinter GUI 窗口
win= tk.Tk()
win.geometry('900x497')
win.config(bg="#ddeeee")
ff=10
#==插入控件
mButton1=dr.DRButton(win, 20, 20, 100, 30, txt=' Run' )
mPlot=dr.DRMonitor(win, 20, 80, 800, 400, wn=4)
#==绑定用户回调函数
mButton1.addCallBackSingle(muserSingle)
#==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。