
Fyplot Documentation

Release 2.0

Vincent Berenz

Oct 11, 2022

Contents

1	Installation	3
2	Source	5
3	Documentation	7

Fyplot is a wrapper over pyqtgraph aiming at making dynamic plotting trivial (at the expense of configurability).

CHAPTER 1

Installation

```
# pip install fyplot
```

After install you may run the demos:

```
# fyplot_demo
```

```
# fyplot_demo2
```


CHAPTER 2

Source

- <https://github.com/intelligent-soft-robots/fyplot/>

CHAPTER 3

Documentation

The full API (1 constructor and 3 methods, really), is covered by the demos:

```
#!/usr/bin/env python

# Copyright (c) 2020 Max Planck Gesellschaft
# Author: Vincent Berenz

from fyplot import function_plot
import math
import time

class A1:
    def __init__(self):
        self.value=0
        self.incr=0.01
    def __call__(self):
        self.value+=self.incr
        return math.cos(self.value)

class A2:
    def __init__(self):
        self.value=0
        self.incr=0.05
    def __call__(self):
        self.value+=self.incr
        return math.sin(self.value)

class B:
    def __init__(self):
        self.value=0
        self.incr=0.005
    def __call__(self):
        self.value+=self.incr
        return math.tanh(self.value)
```

(continues on next page)

(continued from previous page)

```

if __name__ == '__main__':

    title = "function_plot demo"
    windows_size = (800,800)
    period = 50 # period at which the functions will be called

    plot = function_plot.Plot(title,period,windows_size)

    subplot1 = ( (A1(),(255,0,0)), # first plot will get data from A1, and will be red
                 (A2(),(0,255,0)) ) # second plot will get data from A2 and will green
    plot.add_subplot( (-1.0,1.0), # y axis goes between -1.0 and +1.0
                     500, # x axis will display 500 data points (moving window)
                     subplot1 ) # will display values returned by A1 and A2

    subplot2 = ( (B(),(0,0,255)), )
    plot.add_subplot((0.0,1.0),1000,subplot2)

    # starting to plot (non blocking function)
    plot.start()

    # running for 10 seconds
    time_start = time.time()
    while time.time()-time_start<10:
        time.sleep(1)

    # exit
    plot.stop()

```

```

#!/usr/bin/env python

# Copyright(c) 2020 Max Planck Gesellschaft
# Author: Vincent Berenz

from fyplot import dict_plot
import math
import time

if __name__ == '__main__':

    config = dict_plot.Config()

    config.channels = ["L_EB","R_EB","X"]
    config.limits = { "L_EB":(-2,2) ,
                      "R_EB":(-2,2) ,
                      "X":(0,10) }
    config.slots = {"L_EB":["current","target","desired"],
                   "R_EB":["current","target","desired"],
                   "X":["x"]}
    config.slots_colors = {
        "x":(0,255,255),
        "current":(255,0,0) ,
        "target":(255,255,255) ,
        "desired":(150,150,150)
    }

```

(continues on next page)

(continued from previous page)

```
}
config.data_size = 100
config.title = "TEST"
config.windows_size = [1000,1000]

def testing():
    time_start = time.time()
    x = 0.0
    while time.time()-time_start < 5 :
        x+=0.01
        data = {
            "L_EB" : {"current":math.cos(x) , "target":math.sin(x), "desired":0.2,
↪ } ,
            "R_EB" : {"current":math.sin(x) , "target":math.cos(x), "desired":0.2,
↪ } ,
            "X" : {"x":x}
        }
        dict_plot.set_data(data)
        time.sleep(0.005)

dict_plot.start_plotting(config,testing)
```