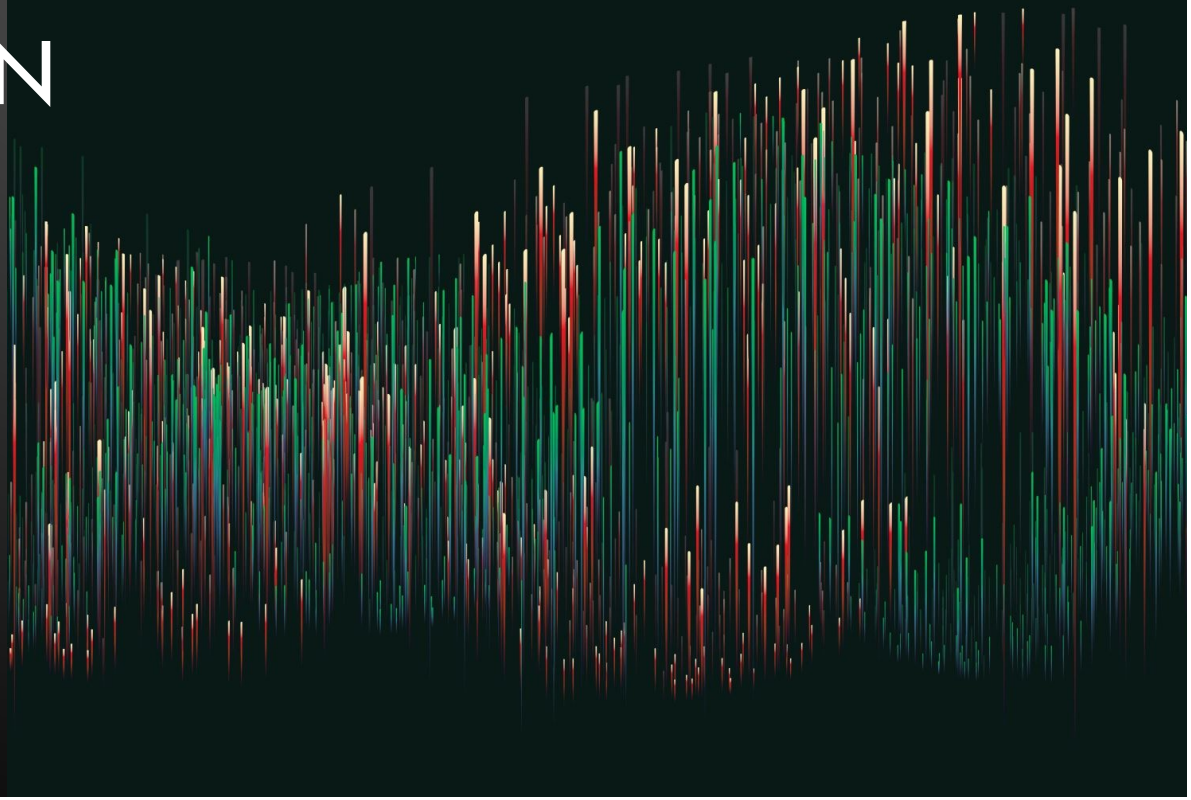
สำหรับการฝึกอบรมเชิงปฏิบัติการ "อุปกรณ์ไอโอทีสำหรับงานควบคุมอุตสาหกรรม"
ภาควิชาฟิสิกส์ คณะวิทยาศาสตร์ ม.นเรศวร 26-27 มิถุนายน 2564

การพัฒนาตัวควบคุม PID โดย MICROPYTHON บน ESP32

DR.VARODOM TOOCHINDA

DEPT. OF MECHANICAL ENGINEERING

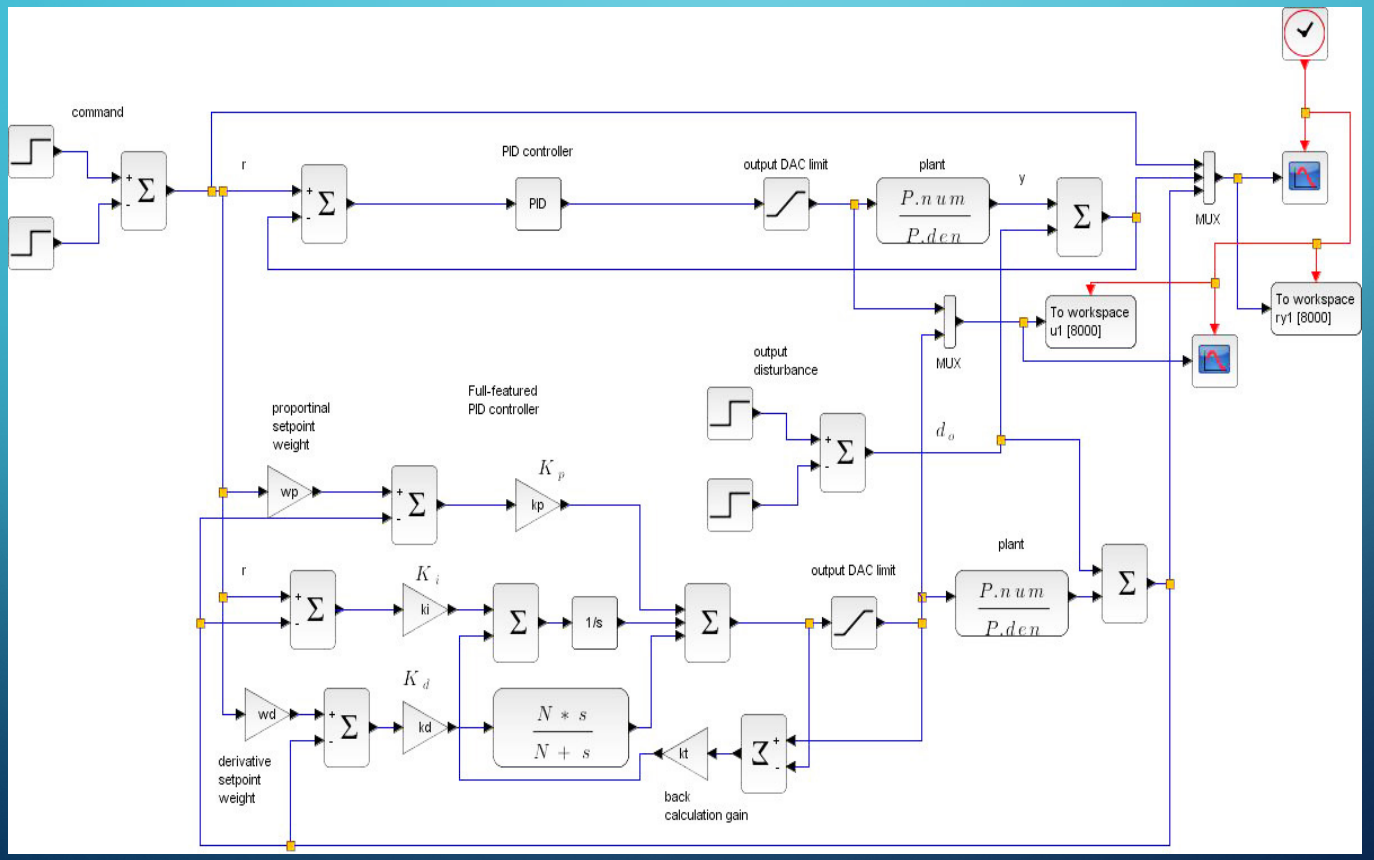
KASETSART UNIVERSITY



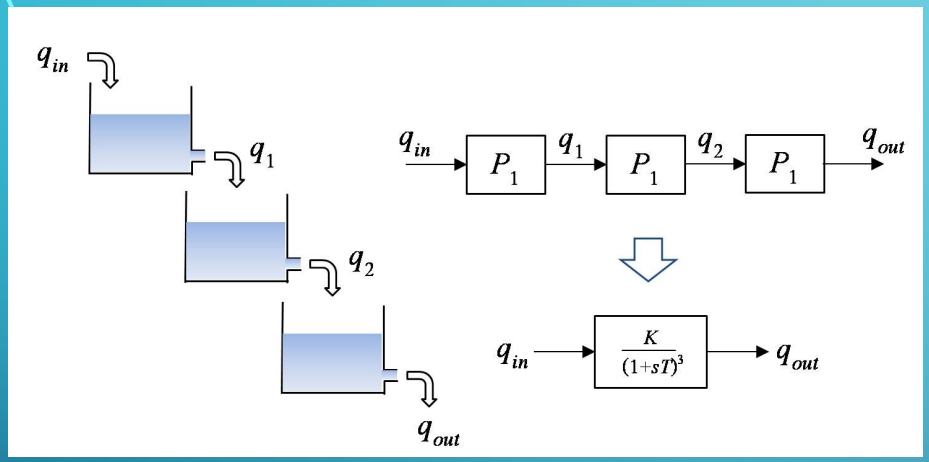
หัวข้อ

- พัฒนาตัวควบคุม PID พร้อมฟังก์ชันเสริม โดย micropython
- สร้าง Freeboard สำหรับควบคุม
- ทดสอบผลตอบสนองขั้นบันได เก็บข้อมูลเปรียบเทียบบน Jupyter notebook
- ปรับค่าพารามิเตอร์ควบคุมและดูผลตอบสนอง

โครงสร้างของตัวควบคุม PID และฟังก์ชันเสริม



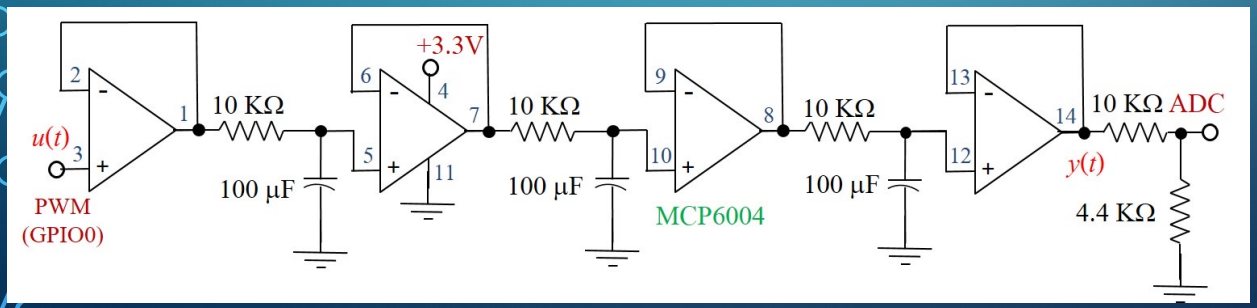
ทดสอบกับบอร์ด LAG3-ESP32 (หรือจำลองพลานต์โดยอัลกอริทึม)

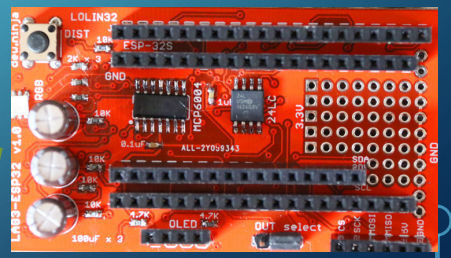


normalized transfer function

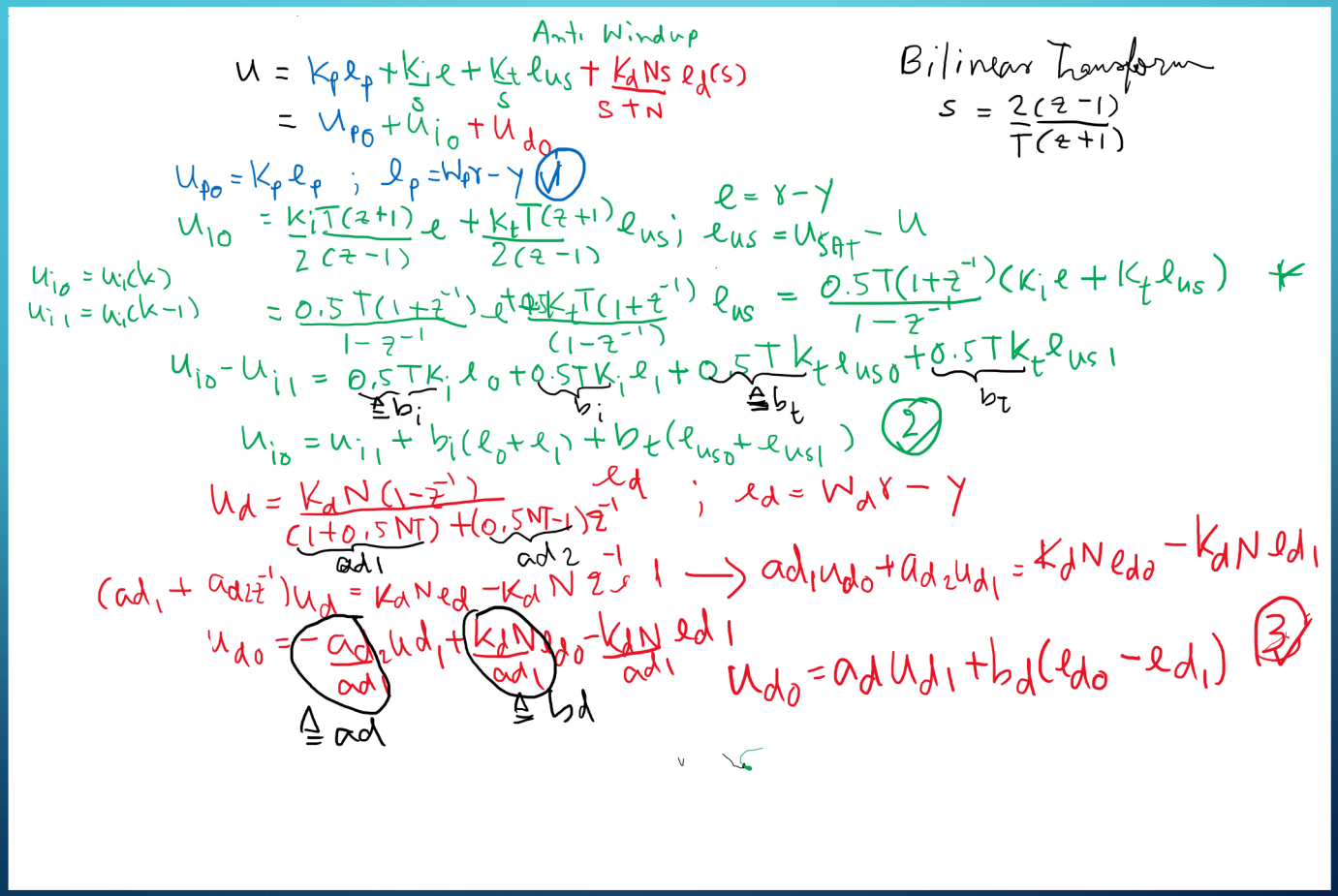
$$\frac{1}{(s+1)^3}$$

Simulated by RC and op-amp circuit





แปลงตัวควบคุมในระบบต่อเนื่องเป็นดีสครีต (1)



แปลงตัวควบคุมในระบบต่อเนื่องเป็นดีสครีต (2)

$$u_0 = u_{p0} + u_{i0} + u_{d0}$$

$$u_{p0} = K_p e_p \quad ; \; e_p = w_p r - y$$

$$u_{i0} = u_{i1} + b_i(e_0 + e_1) + b_t(e_{us0} + e_{us1}) \quad ; \; e = r - y \quad ; \; e_{us} = u_{sat} - u$$

$$u_{d0} = a_{d1} u_{d1} + b_d(e_{d0} - e_{d1}) \quad ; \; e_d = w_d r - y$$

$$b_i = 0.5 T K_i$$

$$b_t = 0.5 T K_t$$

$$a_{d1} = (1 + 0.5NT) \qquad b_d = \frac{K_d N}{a_{d1}}$$

$$a_{d2} = (0.5NT - 1)$$

$$a_d = -\frac{a_{d2}}{a_{d1}}$$

ฟังก์ชันคำนวณสัมประสิทธิ์

```
def PID_update():
    global ad, bd, bi, bt
    bi = 0.5*T*ki
    bt = 0.5*T*kt
    ad1 = 1+0.5*N*T
    ad2 = 0.5*N*T - 1
    ad = -ad2/ad1
    bd = kd*N/ad1
    update_freeboard()
```

ฟังก์ชันตัวควบคุม PID

```
def PID_controller():
    global e1,e0,ed1,ed0, eus1,eus0,
   ui1, ui0, ud1, ud0, u

    # state transfer
    e1 = e0
    ed1 = ed0
    eus1 = eus0

    ui1 = ui0
    ud1 = ud0
    # compute errors for each term
    e0 = r - y
    ep0 = wp*r - y # weighted
          # proportional error
    ed0 = wd*r - y # weighted
          # derivative error
    up0 = kp*ep0 # output of P term
    ui0 = ui1 +bi*(e0+e1) + bt*(eus0+eus1)
# output of I term
    ud0 = ad*ud1 +bd*(ed0 - ed1) # output
of D term
    u = up0 + ui0 + ud0
    u_lim = u
    if u > UMID:
        eus0 = UMID - u  # compute error
             # for back calculation term
        u_lim = UMID          # limit u to
                            # UMID
    elif u < -UMID:
        eus0 = -u - UMID  # compute error
             # for back calculation term
        u_lim = -UMID          # limit u to
                               #-UMID
    u_lim += UMID
    return u_lim
```

การเรียกฟังก์ชันในลูป while True:

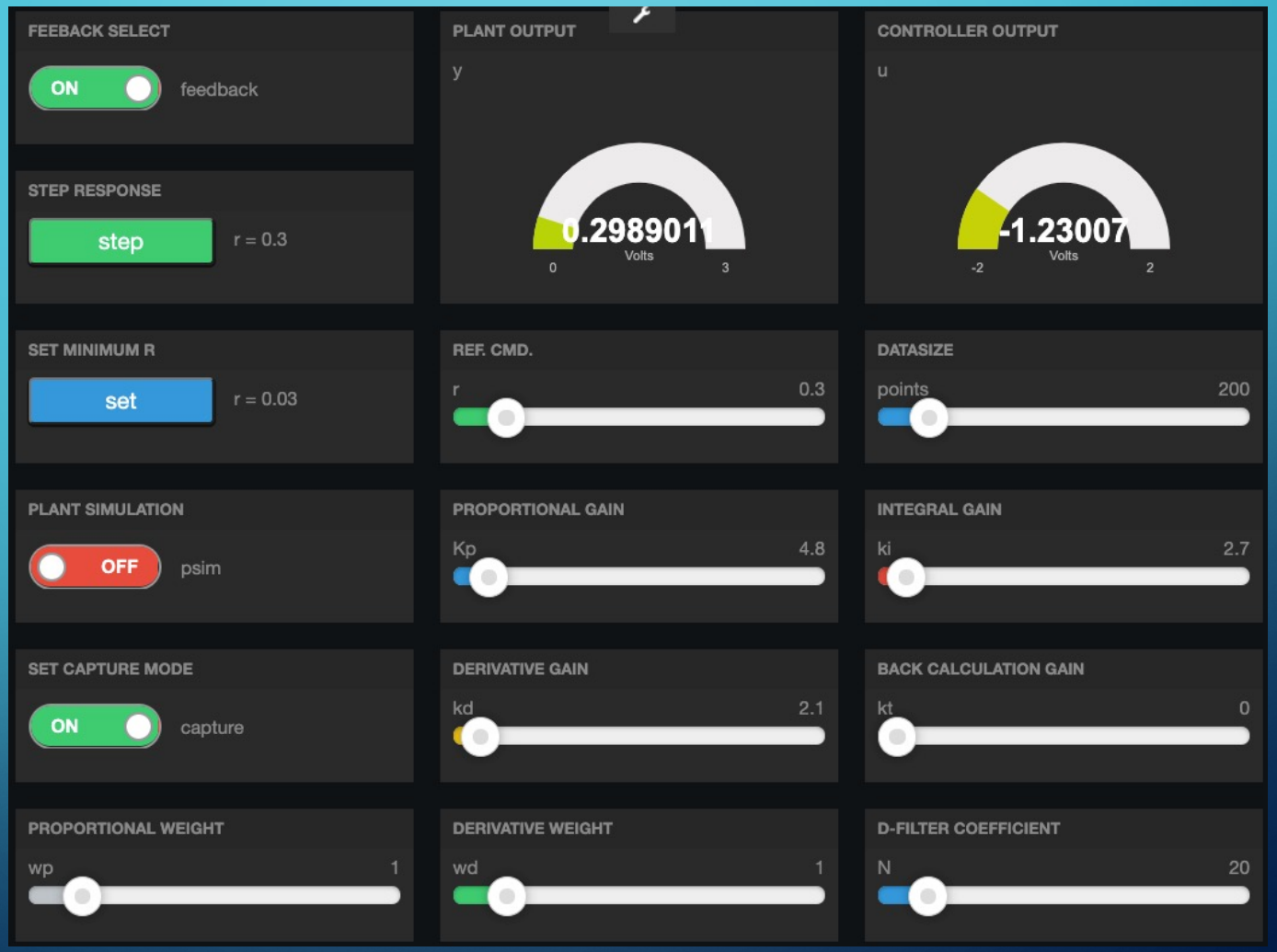
```
if feedback:
    ulim = PID_controller()
else:  # open-loop
    u = r
    ulim = u
u_pwm = int(ulim*v2pwm)
pwm_out.duty(u_pwm)  # output via PWM pin 16
u_dac = int(ulim*v2dac)
dac_out.write(u_dac) # also output to DAC1
```

เพิ่มเงื่อนไขรับคำสั่งใน cmdInt()

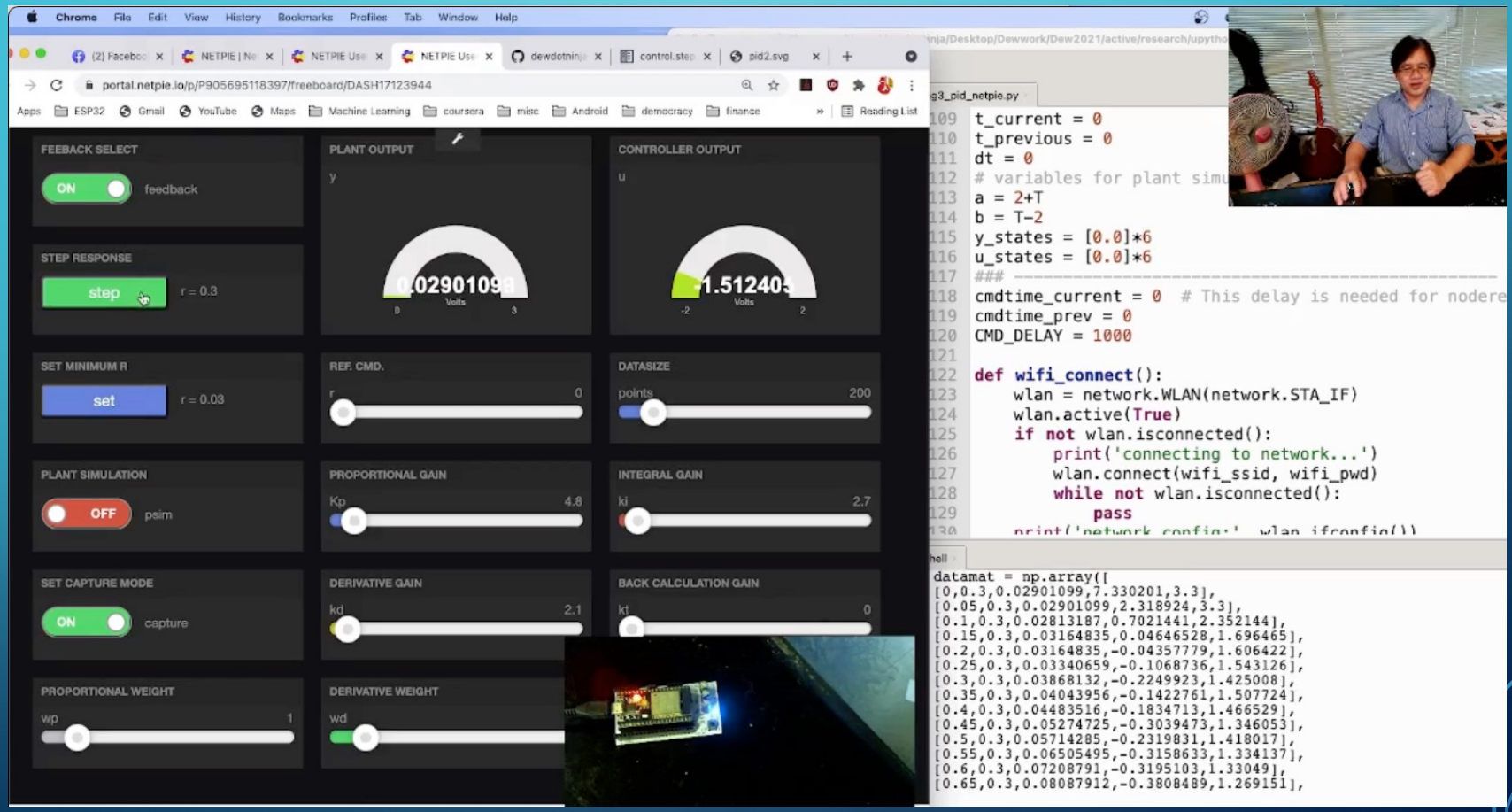
```
        :
    elif cmdstr.lower() == "kp":
            if noparm==1:
                print("Current kp =
    {}".format(kp))
            else:
                kp = float(parmstr)
                if kp > 100: # set maximum kp
                    kp = 100
                elif kp < 0: # minimum kp
                    kp = 0
                PID_update()

        elif cmdstr.lower() == "ki":
            :
```

Freeboard for PID



ทดสอบผลตอบสนองขั้นบันได



คัดลอกใส่ในเซลของ Jupyter notebook

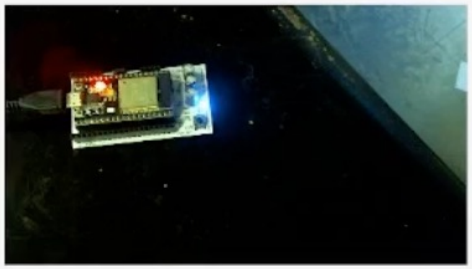
jupyter plot_pid_stepcompare Last Checkpoint: 16 minutes ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3.7 (controlenv)

Run Markdown

In [34]:
```
import numpy as np
import matplotlib.pyplot as plt
import control as ctl
```

In [35]:
```
datamat = np.array([
[0,0.3,0.02901099,7.330201,3.3],
[0.05,0.3,0.02901099,2.318924,3.3],
[0.1,0.3,0.02813187,0.7021441,2.352144],
[0.15,0.3,0.03164835,0.04646528,1.696465],
[0.2,0.3,0.03164835,-0.04357779,1.606422],
[0.25,0.3,0.03340659,-0.1068736,1.543126],
[0.3,0.3,0.03868132,-0.2249923,1.425008],
[0.35,0.3,0.04043956,-0.1422761,1.507724],
[0.4,0.3,0.04483516,-0.1834713,1.466529],
[0.45,0.3,0.05274725,-0.3039473,1.346053],
[0.5,0.3,0.05714285,-0.2319831,1.418017],
[0.55,0.3,0.06505495,-0.3158633,1.334137],
[0.6,0.3,0.07208791,-0.3195103,1.33049],
[0.65,0.3,0.08087912,-0.3808489,1.269151],
[0.7,0.3,0.09054945,-0.4391118,1.210888],
[0.75,0.3,0.1002198,-0.4712322,1.178768],
[0.8,0.3,0.1125275,-0.5822057,1.067794],
[0.85,0.3,0.1230769,-0.584793,1.065207],
[0.9,0.3,0.1362637,-0.6909042,0.9590958],
[0.95,0.3,0.1494505,-0.7547222,0.8952778],
[1.0,0.3,0.1582417,-0.66117,0.98883],
[1.05,0.3,0.1731868,-0.8482413,0.8017587],
[1.1,0.3,0.1846154,-0.8326234,0.8173765],
[1.15,0.3,0.1995604,-0.9700761,0.6799239],
[1.2,0.3,0.2136264,-1.027014,0.6229861],
[1.25,0.3,0.2268132,-1.055505,0.5944954],
[1.3,0.3,0.24,-1.101719,0.5482807],
```



พล็อตเพื่อเปรียบเทียบผลตอบสนองขั้นบันได

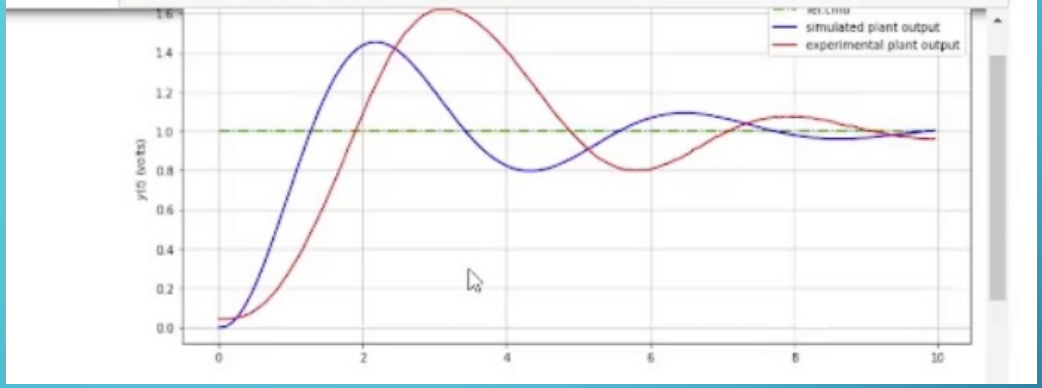
```
In [38]: # use the same parameters as in the ESP32
         kp = 4.8
         ki = 2.74
         kd = 2.1
         pid_compare(kp,ki,kd,datamat,scale=0.3)
```

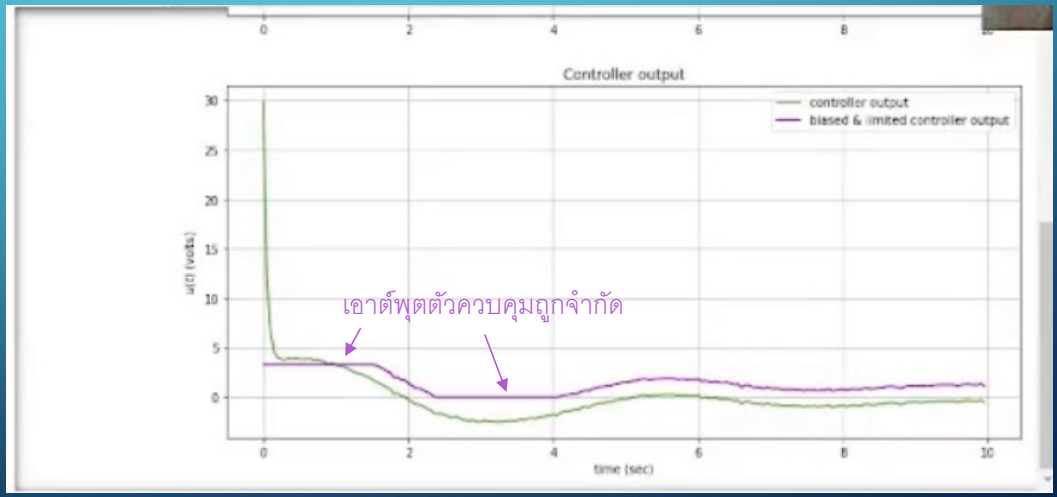
```
In [ ]:
```

```
In [ ]:
```

ผลจากการอิ่มตัว
เมื่อเพิ่มขนาด
ref.cmd = 1.0

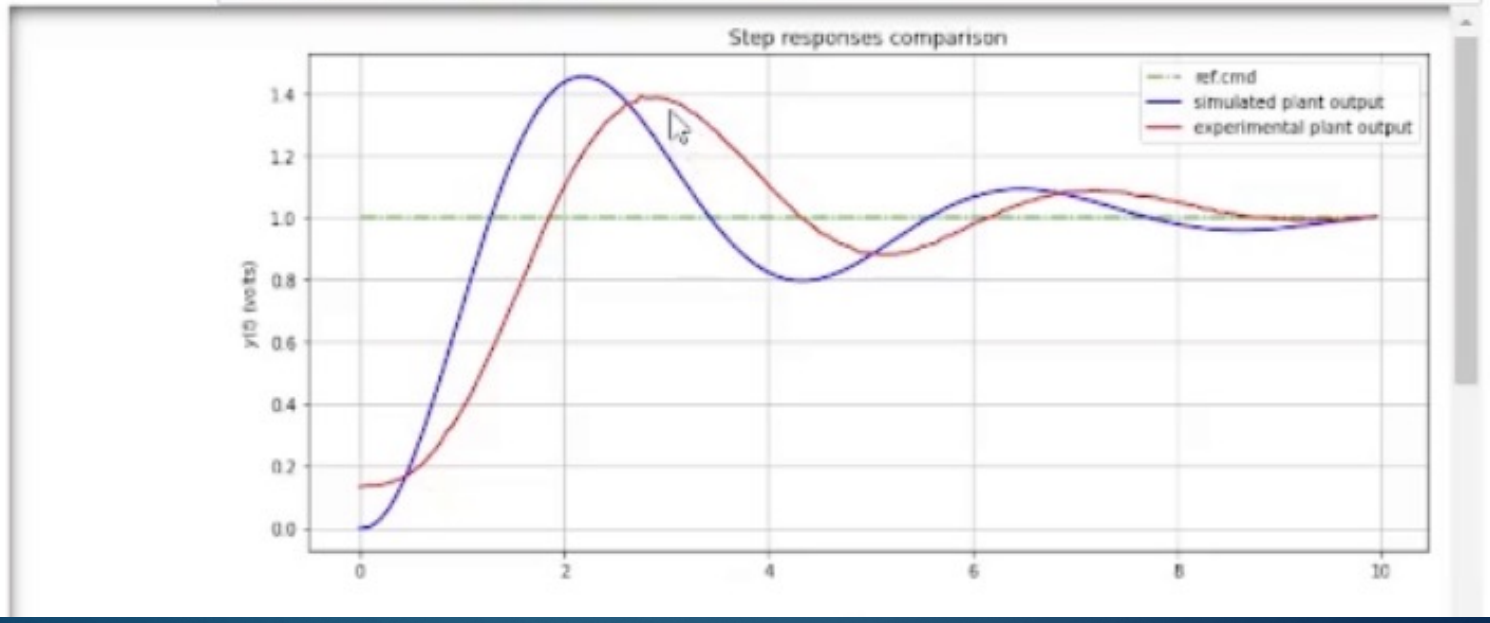
```
In [6]: # use the same parameters as in the ESP32
        kp = 4.8
        ki = 2.74
        kd = 2.1
        pid_compare(kp,ki,kd,datamat,scale=1)
```





เมื่อใช้ back calculation anti-windup (kt=0.5)

```
In [13]: # use the same parameters as in the ESP32
         kp = 4.8
         ki = 2.74
         kd = 2.1
         pid_compare(kp,ki,kd,datamat,scale=1)
```



แบบฝึกหัด

- ทดลองใช้การจำลองพลานต์โดยอัลกอริทึมเปรียบเทียบกับการใช้วงจรอิเล็กทรอนิกส์
- ปรับค่า kp, ki, kd อธิบายผลกระทบต่อผลตอบสนองขั้นบันได
- ทดสอบพารามิเตอร์การให้ค่าน้ำหนักกับพจน์สัดส่วนและอนุพันธ์

Q&A SESSION

Thank You