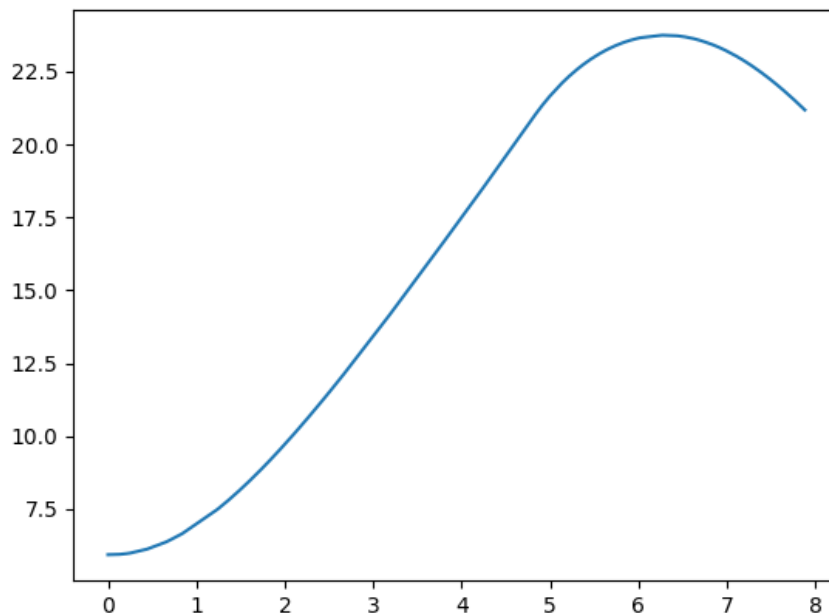


در این تمرین فرض شده است تخمین‌گر وضعیت از قبل پیاده‌سازی شده و در هر لحظه دسترسی کامل به بردار حالت تخمین‌گر داریم. به بیان دیگر مبتنی بر بردار حالت فعلی و اهداف تعریف شده در تمرین، می‌بایست کنترل‌گر طراحی شود. از آنجایی که عملکرد کنترل‌گر می‌بایست در شرایط واقعی ارزیابی شود، می‌بایست کنترل‌گر خود را در تعامل با یک شبیه‌ساز توسعه داده و آن را ارزیابی کنید. بدین منظور نیاز است که به یک شبیه‌ساز برخط متصل شده و برنامه و منطق کنترل‌گر خود را توسعه بدهید. فرآیند این کار در ادامه و در بخش اول تمرین توضیح داده می‌شود.

هدف: کنترل ارتفاع یک هواگرد چهارموتوره توسط یک کنترل‌گر PID

مساله الف: قرار است هواگرد را از سطح زمین به ارتفاع ۲۰ متری ببریم و نگه داریم. توجه شود که در کنترل ارتفاع، هواگرد نباید از ارتفاع ۲۰ متری تجاوز کند (حداکثر تلورانس قابل قبول برای overshoot زدن از ارتفاع ۲۰ متری، ۴۰ سانتی‌متر در نظر گرفته شود) - می‌توانید فرض کنید که بالاتر از ۲۰ متری سقف است و ارتفاع بالاتر منجر به برخورد هواگرد با سقف می‌شود. همچنین زمان اولین مرتبه رسیدن به ارتفاع هدف (با در نظر گرفتن آستانه overshoot) نباید از ۴ ثانیه بیشتر شود.

خروجی ارتفاع مربوط به یک کنترلر بسیار ساده را ملاحظه کنید. در این نمونه حدود ۲ متر از سقف ارتفاع عدول شده است و همچنین سرعت رسیدن به ارتفاع ۲۰ متری بسیار کم می‌باشد (محور افقی زمان بر حسب ثانیه و محور عمودی ارتفاع می‌باشد).



قدم‌های حل:

۱. ابتدا کد را اجرا کنید تا اطلاعاتی از بردار حالت را که برای کنترل نیاز دارید، از هواگرد دریافت کنید و آن را نمایش بدهید.

۲. یک کنترل‌گر PID بر روی خطای موقعیت و خطای سرعت پیاده سازی کنید. دقت کنید که در تمامی مراحل این قسمت از تمرین، چون تنها ارتفاع را کنترل میکنیم، به هر چهار موتور مقدار ورودی مساوی بدهید (دستور هر کدام از موتورها باید مابین ۰ و ۱ باشد)

سعی کنید با تغییر ضرایب P، D و I به رفتار مطلوب خواسته شده برسید. آیا موفق شدید؟ نمودار ارتفاع و فرمان دور موتور را بر حسب زمان ترسیم و تحلیل کنید.

motor command

$$= -(k_p(\text{position error}) + k_d(\text{velocity error}) + k_i(\text{integral of position error}))$$

مساله ب: همان مسئله الف را طوری حل کنید که منحنی سینوسی زیر را در ارتفاع (یعنی منهای Z) متابعت کند. خروجی سیگنال کنترلی و ارتفاع لحظه‌ای را بر حسب زمان رسم و تحلیل کنید.

$$h = 15 + 5 \sin(0.4\pi t)$$

توجه داشته باشید که در این قسمت، به دلیل منحنی دار بودن رفتار دلخواه، رابطه کنترل‌گر می‌تواند مؤلفه‌های دیگری نیز داشته باشد. برای اطلاعات بیشتر به جزوه درس مراجعه کنید.

بلوک‌های اصلی کد:

برای سهولت، یک بسته پایتونی `tools` در پیوست تمرین در اختیار شما قرار گرفته است و نمونه‌ای از بلوک‌های حلقه کنترلی مورد نیاز در ادامه آمده است^۱. تنها بسته مورد نیاز^۲ ماژول `tools`، بسته `redis-py` است که می‌توانید با روش‌های مختلف نظیر استفاده از نصاب `pip` آن را دریافت و نصب کنید:

```
pip install redis
```

هر یک از دانشجویان دارای یک شناسه هواگرد و رمز ورود هستند که جداگانه (در ماژول `credentials.py`) در اختیار آنها قرار می‌گیرد. در ادامه بخش‌های اصلی یک بلوک کنترلی توضیح داده می‌شود.

```
import numpy as np
from tools.sim_tools import SimConnector
from tools.viewer import LiveViewer
import time
from tools.credentials import robot_id, password

sim = SimConnector(robot_id, password)
sim.reset()
viewer = LiveViewer(sim, 3)
viewer.start_view()
data_log = np.zeros([1e5, 2])
cntr = 0
# your initializations here
control_rate = 20.0 # Hz
control_period = 1.0/control_rate
while True:
    state = sim.get_drone_state()
    height = -state['position'][2] # is needed for control
    data_log[cntr, 0] = height # for logging and plotting purpose

    # your control logic here
    # calculate motor_pwm
    # motor_pwm
    data_log[cntr, 1] = motor_pwm

    cntr += 1
    sim.send_motor_command(motor_pwm, motor_pwm, motor_pwm, motor_pwm, control_period)
    time.sleep(control_period)

    # termination criteria
    if termination criteria satisfied:
        break

data_log = data_log[:cntr, :] # remove unnecessary memory allocs

# plottings here using data_log
```

^۱ - فایل zip پیوست تمرین از آدرس http://silo.aipark.ir/amb2024/assignments/04/codes_and_tools.zip قابل دریافت است.

^۲ - البته به سایر بسته‌های مرسوم نظیر `numpy`، `scipy`، `matplotlib` و `opencv-python` نیز ممکن است نیاز پیدا کنید.

۱. ابتدا کلاس `SimConnector`، با شناسه هواگرد و رمز ورود ایجاد می‌شود.
۲. با تابع `reset` می‌توانید هواگرد را در خانه قرار بدهید.
۳. ابزارک `LiveViewer` با استفاده از تابع `sim.get_cam_image` نمونه‌ای از کلاس `SimConnector` می‌تواند با نرخ مشخص به صورت زنده تصویر دوربین هواگرد را به شما نشان بدهد (نرخ ۳ هرتز در کد نمونه قرار داده شده است – باتوجه به شرایط اینترنت می‌توانید آن را بیشتر کنید).
۴. با استفاده از تابع `get_drone_state` اطلاعات حالت لحظه‌ای هواگرد دریافت می‌شود.
۵. در بخش `control logic` می‌بایست با توجه به مقادیر بردار حالت و هدف ارتفاع که در نظر دارید مقادیر دور موتورهای جلو-راست، عقب-چپ، جلو-چپ و عقب-راست را به همراه مدت زمان اعمال فرمان کنترل محاسبه کنید و توسط تابع `send_motor_command` فرمان بدهید. برای مدت زمان می‌توانید از مقدار `1.0/control_rate` (یعنی مدت زمان طول کشیدن هر مرتبه از حلقه) استفاده کنید.
۶. در صورتی که به هدف ارتفاع رسیده‌اید می‌توانید از حلقه کنترل خارج شوید.
۷. نرخ مطلوب کنترل در تمرین را می‌توانید ۲۰ هرتز در نظر بگیرید.
۸. در صورتی که هواگرد شما در پرواز دچار اختلال غیر قابل بازگشت شد، می‌توانید از دستور `reset` برای بازگرداندن آن به سطح زمین استفاده کنید. همچنین پس از آنکه به ارتفاع رسیدید و مجدد خواستید هواگرد را از زمین بلند کنید می‌توانید با همین دستور هواگرد را روی سطح زمین نشاند و مجدد کنترل را آغاز کنید.

چند نکته:

- در مسائل پروازی معمولاً دستگاه مختصات `NED` بیشتر از `ENU` استفاده می‌شود. در نتیجه هرچه هواگرد از زمین فاصله بیشتری بگیرد، موقعیت آن در محور `Z`، کمتر می‌شود (جهت مثبت محور `Z` رو به پایین است).
- کلیدهای دیکشنری حالت با استفاده از نام آن‌ها قابل درک هستند. اغلب مقادیر دارای لیست‌های سه‌گانه `x`، `y` و `Z` می‌باشند؛ به جز `orientation` که در قالب کلاس `Rotation` از `Scipy` بازنمایی می‌شود.
- مقدار مربوط به زمان شبیه‌ساز تحت عنوان `ts` است که زمان برحسب نانو ثانیه می‌باشد. مقدار زمان‌ها را با کم کردن زمان‌ها از اولین `ts` نرمالیزه از صفر کنید و در نمودارها بر حسب ثانیه یا میلی‌ثانیه ترسیم نمایید.
- اتصال اینترنت شما به سرور شبیه‌ساز می‌بایست اتصال با ثبات و با تاخیر کم باشد. برای بررسی آن از دستور زیر استفاده کنید:

```
ping www2.aipark.ir -t
```

نمونه‌ای ایده‌آل از زمان ارسال بسته `icmp` و مدت زمان ارسال و بازگشت آن در تصویر زیر آمده است:

```
Pinging www2.aipark.ir [46.102.130.168] with 32 bytes of data:
Reply from 46.102.130.168: bytes=32 time=1ms TTL=128
Reply from 46.102.130.168: bytes=32 time=1ms TTL=128
Reply from 46.102.130.168: bytes=32 time=1ms TTL=128
Reply from 46.102.130.168: bytes=32 time=1ms TTL=128
Reply from 46.102.130.168: bytes=32 time=1ms TTL=128
Reply from 46.102.130.168: bytes=32 time=1ms TTL=128
```

اگر شرایط مناسب نبود، بهتر است اینترنت را تغییر بدهید یا در ساعاتی از روز تمرین را انجام بدهید که شرایط اینترنت مناسب تر باشد. نمونه زیر نیز می تواند قابل قبول باشد:

```
PING www2.aipark.ir (46.102.130.168) 56(84) bytes of data:
64 bytes from 46.102.130.168 (46.102.130.168): icmp_seq=1 ttl=57 time=5.20 ms
64 bytes from 46.102.130.168 (46.102.130.168): icmp_seq=2 ttl=57 time=5.72 ms
64 bytes from 46.102.130.168 (46.102.130.168): icmp_seq=3 ttl=57 time=5.20 ms
64 bytes from 46.102.130.168 (46.102.130.168): icmp_seq=4 ttl=57 time=4.79 ms
64 bytes from 46.102.130.168 (46.102.130.168): icmp_seq=5 ttl=57 time=5.17 ms
64 bytes from 46.102.130.168 (46.102.130.168): icmp_seq=6 ttl=57 time=4.96 ms
64 bytes from 46.102.130.168 (46.102.130.168): icmp_seq=7 ttl=57 time=5.49 ms
64 bytes from 46.102.130.168 (46.102.130.168): icmp_seq=8 ttl=57 time=5.42 ms
```

- در صورتی که ping time شما کمتر از یک میلی ثانیه بوده و این میزان jitter نداشته باشد، شرایط شبکه ایده آل است. در صورتی که این میزان از ۱۰ میلی ثانیه بیشتر باشد یا نوسان زیادی داشته باشد، تلاش کنید اینترنت خود را تغییر بدهید تا شرایط بهبود یابد.