ANSWERS

U6.1 Two sources, one state

State estimation · University · about 25 min

What this lesson is about

A signal that drifts and a signal that is noisy, and why either alone is worse than both.

- The complementary filter
- Choosing a
- Wrapping
- Seeing all three

Questions

6 marks in all

1. A complementary filter uses a = 0.98 and dt = 0.1 s. What is its time constant in seconds? [1 mark]

> **Answer:** 4.9 (accept within 0.01). a x dt / (1 - a) = 0.98 x 0.1 / 0.02 = 4.9 s, so drift is pulled out over about 5 seconds.

2. Blending 359 degrees with 1 degree, first naively and then with the wrap. What does it print? [1 mark]

```
mine, fused = 359.0, 1.0
print((mine + fused) / 2)
f = fused
while f - mine > 180:
    f -= 360
while mine - f > 180:
    f += 360
print(f, ((mine + f) / 2) % 360)
```

> **Answer:**
>
> ```
> 180.0
> 361.0 0.0
> ```
>
> The naive average is 180, pointing the wrong way. Unwrapped, 1 becomes 361, within half a turn of 359, and the blend is 360, which is 0: correct.

3. Two ticks of the complementary filter. What does it print? [1 mark]

```
A, DT = 0.98, 0.1
mine = 90.0
for rate, fused in ((10.0, 95.0), (10.0, 96.0)):
    mine = A * (mine + rate * DT) + (1 - A) * fused
    print(round(mine, 3))
```

> **Answer:**
>
> ```
> 91.08
> 92.158
> ```
>
> Tick 1: 0.98 x (90 + 1) + 0.02 x 95 = 89.18 + 1.9 = 91.08. Tick 2: 0.98 x 92.08 + 0.02 x 96 = 92.158. The gyro carries the change and the fused heading nudges it.

4. The filter is run with a = 0.7. What goes wrong? [1 mark]

- ☐ **A** The noise on the fused heading comes straight through
- ☐ **B** Gyro drift is never corrected
- ☐ **C** The heading wraps at 180
- ☐ **D** The filter becomes unstable and diverges

> **Answer:** A. a = 0.7 gives the fused heading 30 percent of each update, a time constant of about 0.23 s, so its wobble dominates. Too close to 1 is the opposite failure: drift is corrected too slowly.

5. Which statements about the two heading sources are correct? [1 mark]

Tick every answer that is true.

- ☐ **A** The integrated gyro is smooth but drifts without limit
- ☐ **B** imu()[0] is noisy but does not run away
- ☐ **C** The integrated gyro is noisy but does not drift
- ☐ **D** imu()[0] is smooth but drifts without limit

> **Answer:** A, B. One drifts and does not wobble. The other wobbles, and wanders a few degrees, but does not run away. The complementary filter uses each where it is good.

6. In frequency terms, what does the complementary filter do to each source? [1 mark]

- ☐ **A** High pass on the gyro, low pass on the absolute heading, summing to unity gain
- ☐ **B** Low pass on the gyro, high pass on the absolute heading
- ☐ **C** Low pass on both
- ☐ **D** A band pass on the gyro only

> **Answer:** A. Short-term changes come from the gyro and long-term level from the absolute reading. The two weights add to one at every frequency, hence complementary.

The task: a complementary filter

Turn the robot about, running a complementary filter on the heading, plotting `gyro`, `fused` and `mine`, and print `my heading:` at the end. `heading()` is the truth and is not allowed, so plot the three headings themselves. They will lie almost on top of each other, which is expected: they differ by a few degrees on a chart that runs to 360.

```
from bugbot import *
connect()

DT = 0.1
A = 0.98
mine = imu()[0]
```

The hint students can ask for: Integrate imu()[1] for a heading that is smooth but drifts, and read imu()[0] for one that is noisy but does not. mine = 0.98 * (mine + rate * dt) + 0.02 * fused, every tick.

A solution

```
from bugbot import *
connect()

DT = 0.1
A = 0.98
gyro = 0.0
mine = imu()[0]

def step(n=1):
    global gyro, mine
    for i in range(n):
        rate, fused = imu()[1], imu()[0]
        gyro += rate * DT
        # keep the two within half a turn of each other before blending them
        while fused - mine > 180:
            fused -= 360
        while mine - fused > 180:
            fused += 360
        mine = A * (mine + rate * DT) + (1 - A) * fused
        plot("gyro", gyro % 360)
        plot("fused", imu()[0])
        plot("mine", mine % 360)
        wait(DT)

drive(0, 0, 45)
step(60)
stop()
step(10)
drive(0, 0, -45)
step(30)
stop()
step(10)
print("my heading:", round(mine % 360, 1))
```

Any program that meets the task's checks is marked correct in the simulator; this is one way, not the only way.