



U1.5 Seeing the signal

What this lesson is about

plot(): reading a control loop as a chart instead of a column of numbers.

- What to plot
- Reading the shapes
- Plotting truth against belief

Questions

7 marks in all

1. Which three lines, plotted together, answer most questions about a feedback loop?

[1 mark]

Tick every answer that is true.

- ☐ A The measurement
- ☐ B The error, target minus measurement
- ☐ C The command you sent
- ☐ D The battery voltage
- ☐ E The loop period

Answer: A, B, C. The measurement says what the robot thinks is happening, the error says whether the loop is winning, and the command shows whether you are asking for more than the hardware can give.

2. The error stays large while the plotted command sits flat at 100. What is happening?

[1 mark]

- ☐ A The command is saturated, and no amount of extra gain will fix it
- ☐ B The gain is too low and should be raised
- ☐ C There is a steady state error, which needs an integral term
- ☐ D There is too much delay in the loop

Answer: A. A command pinned at its limit means the hardware is already giving everything it has. More gain or an integral term only asks harder for what cannot be delivered.

3. The error settles smoothly, but to a steady value that is not zero. What does that shape mean?

[1 mark]

- ☐ A A steady state error, which needs an integral term
- ☐ B Under-damped: the gain is a little high
- ☐ C The sign of the correction is wrong
- ☐ D Sensor noise is reaching the motors

Answer: A. Settling somewhere other than zero is steady state error. The integral term in U5 is what removes it.

4. The error overshoots zero, comes back, and overshoots again by less each time. What does that shape mean? [1 mark]

- ☐ A Under-damped: the gain is a little high, or there is some delay in the loop
- ☐ B The correction has the wrong sign
- ☐ C A steady state error
- ☐ D The loop is saturated

Answer: A. Decaying overshoot is under-damping. A wrong sign or far too much gain would make the error grow instead.

5. The plotted error gets bigger on every tick. What are the likely causes? [1 mark]

- ☐ A The correction has the wrong sign, or the gain is far too high
- ☐ B A steady state error that needs an integral term
- ☐ C Sensor noise reaching the motors
- ☐ D The loop is running too fast

Answer: A. A growing error means the loop is making things worse: either it pushes the wrong way, or it pushes so hard it overshoots further each time.

6. A chart shows the error as a fuzzy band around zero, and the rotation command jitters along with it. What is reaching the motors? [1 mark]

Answer: sensor noise. Noise on the measurement passes straight through the gain into the command. Filtering it, in U4, is the fix.

7. A plot of believed y from odometry() and true y from position() shows two lines that start together and slowly separate. What does that picture show? [1 mark]

- ☐ A Dead reckoning drifting away from the truth, which state estimation exists to prevent
- ☐ B A controller that is oscillating
- ☐ C A command that has saturated
- ☐ D The lag of the drive after a step

Answer: A. The robot's belief accumulates error while the truth does not. Keeping those two lines together is the job of the estimators in U6 and U7.

The task: plot the error

Drive into the green zone under proportional control, plotting two lines as you go: `error` and `rotation` .

```
from bugbot import *
connect()

def wrapped(a):
    return (a + 180) % 360 - 180
```

The hint students can ask for: `plot(name, value)` once per loop, next to where you work the value out. Two names, two lines.

A solution

```
from bugbot import *
connect()

def wrapped(a):
    return (a + 180) % 360 - 180

while position()[1] < 120:
    error = wrapped(0 - heading())
    rotation = error * 3
    plot("error", error)
    plot("rotation", rotation)
    drive(70, 0, rotation)
    wait(0.1)
stop()
```

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