

Opmerking: als het mogelijk is voor bezemers, maak ik graag gebruik van de extra 2 weken om mijn systeem obv de tests te verbeteren.

Testen

Het uiteindelijke product wijkt af van het testplan omdat ik de opdracht toen anders voor had gesteld. Maar de gekozen soort testen zijn onveranderd.

Unit-testen

Test cases

In de volgende bestanden zijn unit-tests en bijbehorende test-cases te vinden:

1. src/actuators/LinearActuator.cpp
2. src/sensors/unitTests.cpp
3. tests/unit_tests.py

Resultaten

Niet heel geavanceerde maar essentiële tests, omdat de sensoren later in het systeem worden gebruikt en betrouwbaar moeten zijn in hun werking

Bovendien zijn de unit-testjes tussen C++ en Python belangrijk om te controleren of de koppeling daadwerkelijk functioneert:

1. De eerste test leverde een interessant resultaat op voor de omrekening van inch naar cm:

```
LinearActuator.cpp:88: FAILED:
  REQUIRE( actuator.getPositionCM() == 10.0 )
with expansion:
  9.99998 == 10.0
```

Waarschijnlijk is dit te wijten aan het gebruik van simpele doubles voor de data. Aangezien het voor deze use case niet nodig is om tot op de 0.00001 nauwkeurig te zijn, beschouw ik deze test als geslaagd. Overigens waren de overige tests succesvol.

```
=====
All tests passed (4 assertions in 2 test cases)
```

- 2.
3. Door de koppeling van C++ en Python treedt ongeveer hetzelfde probleem als test 1 op: de waarden wijken enigszins af (opgegeven waarden 20.0 en 50.0):

```
19.953917273889452
49.67071334629697
3.9645665612695677
```

maar omdat ook hier de resultaten nauwkeurig genoeg zijn, keur ik ze goed.

Integratietest

Test case

Controleren of de waterpomp in- en uitschakelt zoals verwacht op basis van de gemeten bodemvochtigheid. (test van regel 104 t/m 107 in regelsysteem.py)

Uitslag

Het raam reageert juist op de waarden en gegeven setpoint.

```
sensorwaarden: soil_sensor: 704, temperatuursensor: 18.97, luchtvochtigheidssensor: 49.14
raam is dicht
sensorwaarden: soil_sensor: 852, temperatuursensor: 19.14, luchtvochtigheidssensor: 48.53
raam is dicht
sensorwaarden: soil_sensor: 865, temperatuursensor: 19.11, luchtvochtigheidssensor: 46.87
raam is dicht
sensorwaarden: soil_sensor: 850, temperatuursensor: 18.82, luchtvochtigheidssensor: 47.8
raam is dicht
```

```
sensorwaarden: soil_sensor: 871, temperatuursensor: 23.56, luchtvochtigheidssensor: 51.19
raam is open
sensorwaarden: soil_sensor: 867, temperatuursensor: 23.63, luchtvochtigheidssensor: 54.14
raam is open
sensorwaarden: soil_sensor: 882, temperatuursensor: 23.84, luchtvochtigheidssensor: 53.28
raam is open
sensorwaarden: soil_sensor: 855, temperatuursensor: 23.38, luchtvochtigheidssensor: 51.47
raam is open
```

Echter, tijdens de test is gebleken dat de simulatie niet goed lijkt te werken en de waarde niet snel genoeg verlaagt of verhoogt. Dit kan betekenen dat het simulatiegedeelte (vanaf regel 109 in regelsysteem.py) van het regelsysteem nog aangepast moet worden om beter te kunnen reageren op veranderingen in de bodemvochtigheid. Het is waarschijnlijk dat er aanvullende berekeningen of algoritmen nodig zijn om de simulatie sneller en beter te maken.

Systeem-/blackboxtesten

Test cases

1. Laat de control_loop() een paar minuten draaien.
2. Testen of de @time_exec_logger decorator functie op de control_loop() werkt

Uitslagen

1. De control_loop() start op zonder foutcodes. En lijkt ook constant op de gegeven interval te updaten:

```
D:\Users\eikel\Documents\Onderwijs\School\ATP\Eindopdracht\simulatie\tests>python regelsysteem.py
sensorwaarden: soil_sensor: 892, temperatuursensor: 21.66, luchtvochtigheidssensor: 54.09
sensorwaarden: soil_sensor: 857, temperatuursensor: 21.82, luchtvochtigheidssensor: 54.18
sensorwaarden: soil_sensor: 855, temperatuursensor: 21.84, luchtvochtigheidssensor: 54.49
sensorwaarden: soil_sensor: 890, temperatuursensor: 21.65, luchtvochtigheidssensor: 53.53
sensorwaarden: soil_sensor: 1006, temperatuursensor: 22.08, luchtvochtigheidssensor: 52.85
sensorwaarden: soil_sensor: 895, temperatuursensor: 22.01, luchtvochtigheidssensor: 53.65
sensorwaarden: soil_sensor: 894, temperatuursensor: 21.93, luchtvochtigheidssensor: 52.03
sensorwaarden: soil_sensor: 438, temperatuursensor: 21.57, luchtvochtigheidssensor: 54.29
sensorwaarden: soil_sensor: 869, temperatuursensor: 22.14, luchtvochtigheidssensor: 52.75
sensorwaarden: soil_sensor: 680, temperatuursensor: 21.79, luchtvochtigheidssensor: 53.88
```

2. Tijdens het testen kwam ik achter een rekenfout. Na correctie werkt de decorator nu wel naar behoren:

```
Functie control_loop werd aangeroepen op 2023-04-12 01:22 en duurde 1.0000534057617188 seconden.
Functie control_loop werd aangeroepen op 2023-04-12 01:22 en duurde 1.0111253261566162 seconden.
Functie control_loop werd aangeroepen op 2023-04-12 01:22 en duurde 1.0089399814605713 seconden.
Functie control_loop werd aangeroepen op 2023-04-12 01:22 en duurde 1.007404088973999 seconden.
Functie control_loop werd aangeroepen op 2023-04-12 01:22 en duurde 1.014103889465332 seconden.
Functie control_loop werd aangeroepen op 2023-04-12 01:22 en duurde 1.0154387950897217 seconden.
Functie control_loop werd aangeroepen op 2023-04-12 01:22 en duurde 1.0099472999572754 seconden.
Functie control_loop werd aangeroepen op 2023-04-12 01:22 en duurde 1.013939380645752 seconden.
Functie control_loop werd aangeroepen op 2023-04-12 01:22 en duurde 1.0125131607055664 seconden.
Functie control_loop werd aangeroepen op 2023-04-12 01:22 en duurde 1.001906394958496 seconden.
Functie control_loop werd aangeroepen op 2023-04-12 01:22 en duurde 1.0109446048736572 seconden.
```

Criteria van de opdracht

- Implementatie: onderaan te vinden in tests/regelsysteem.py (*waarschuwing: grote code-base*)
- Aspect-oriented decorator: te vinden in tests/regelsysteem.py onder de naam 'time_exec_logger' (regel 19)
- Python-C++ binding: er zijn twee sensoren (te vinden in src/sensors) en twee actuators (src/actuators)
- Simulator: te vinden in tests/CLI-simulator.py