Digital Twin for Plant Health Monitoring

(Slide 2) Motivation and Problem Statement

Problem Statement: Given a physical plant, develop a digital twin to monitor health, predict needs and provide recommendations and actions.

Digital Twins appear in a context **Industry 4.0** as a way to model and simulate physical systems in a digital environment.

Industry 4.0 The fourth industrial revolution is the application of **advanced** digital technologies to transform **HOW** industrial companies operate, aiming to boost efficiency, agility and quality.

It is characterized by the integration of **IoT**, **automation**, **robotics**, **big data**, …

In the context of **Plant Health Monitoring**, a digital twin can be used to

- **Improve decision making, prediction**: Digital twins integrate data from multiple sources to provide a comprehensive view of the system which can be analyzed to make better decisions and predictions e.g. **optimize** irrigation, fertilization, prevent diseases, …
- **Constant monitoring**: Digital twins allow to monitors systems in real-time which is essential with constant changes in the environment.
- **Reduce costs**: By simulating the system in a digital environment, it is possible to optimize the system and reduce costs.
- **Reduce human error**: By automating the monitoring and decision making process, it is possible to reduce human error.
- …

(Slide 3) Formal Problem Description

Given a physical plant P evolving in a continuously varying environment E. The goal is to develop a digital twin D which is a dynamic, real-time virtual replica of of the plant's environmental context and state.

Let E be characterized by a set of environmental variables, at any time t

$$E(t) = \{L(t), M(t), T(t), H(t)\}$$

[1] where :
- $L(t) \in \mathbb{R}^+$: Light intensity measured in lux
- $M(t) \in [0, 100]$: Soil moisture measured in percentage
- $T(t) \in R$: Temperature measured in degrees Celsius
- $H(t) \in [0, 100]$: Humidity measured in percentage

can be measured with their respective sensors.

The plant P is characterized by its internal state $I(t)$ which can be described by a set of internal variables. At any time t, $I(t) = \{G(t), C(t)\}$ where:
- $G(t) \in [0, 100]$: Growth stage of the plant measured in percentage
- $C(t) \in [0, 255]^3$: Color of the plant measured in RGB values.

---

[1] This is a simplification, in practice, the environment can be characterized by many more variables.

The digital twin D is a model of the plant P and its environment E that not only let us visualize the state of the plant and the factors influencing it, but also predict its future state and provide recommendations and actions to optimize its health.

To formalize the previous description, we can define the state of the system $S(t)$ as a function of the environment $E(t)$ and and the plant's internal state $I(t)$. We can write $S(t) : E(t) \times I(t)$

(Slide 4) System Design and Implementation

The digital twin system is composed of the following sensors:

- Temperature & Humidity Sensor DHT22 -> GPIO
  - DHT22 is more accurate than DHT11
  - Measures temperature from −40 to 80°C (+- 0.5°C error)
  - Measures humidity from 0 to 100% (+- 2% error)
  - This sensor will allow us keep an optimal temperature and humidity for the plant. (healthy environment)
- Light Intensity Sensor (Adafruit BH1750 -> I2C
  - Measures light intensity in lux
  - Can measure from 0 to 65k+ lux
  - This sensor will allow us to keep an optimal light intensity for the plant. (photosynthesis)
- Soil Moisture Sensor Adafruit STEMMA -> I2C
  - Capacitive because resistive sensors will oxidize over time
  - Give results from 200 (very dry) to 2000 (very wet) as well as temperature (+- 2°C error)
  - This sensor will allow us to keep an optimal soil moisture for the plant. (watering)

All the sensors can be used with their respectives adafruit libraries in Python which makes it easy to integrate them in the system. The DHT22 sensor use the regular GPIO protocol, while the BH1750 and STEMMA sensors use the I2C protocol which are both wrapped in the adafruit libraries.

The data from the sensors is sent to the digital twin system which is implemented in Python using the **MQTT** protocol. The system is composed of the following components:

- **MQTT Broker**: The broker is responsible for receiving the data from the sensors and sending it to the digital twin.
- **Digital Twin**: The digital twin is responsible for receiving the data from the sensors, updating the state of the plant, storing the data and providing recommendations and actions.
- **Dashboard**: The dashboard is a web interface that displays the state of the plant, the environment and the recommendations. It is implemented using Flask and matplotlib for the real-time plots.

(Slide 5) Example Sensor: Soil Moisture

The main focus in the following days/weeks will be to integrate the soil moisture sensor in the Digital Twin system as it is the most **controllable sensor** which is perfect for a first implementation. It will allow us to test the system and make sure that the data flow is working correctly. Once the soil moisture sensor is integrated, the others shouldn't be too hard to integrate.

⇒ Usage of mock data while waiting for the sensor to arrive.

(Slide 6) Data Flow: Soil Moisture Example Pipeline

Sensor ⟶ Raspberry Pi (GPIO Pins) ⟶ Python Script (processing) ⟶ Database ⟶ Dashboard.

(Slide 7) Status and Challenges

Status:
- First reading of the documentation
- Started the implementation of the system architecture
- Trying to mock the data while waiting for the sensor to arrive

Challenge:
- Integrating MQTT in the system

Future Step:
- Integrate the soil moisture sensor in the system
- Check the data flow from the sensor to the dashboard