PLC-controlled LED Lighting Gantry Robot

Lacey Hunt, Dr. John C. Shovic

Department of Computer Science, University of Idaho

Introduction

- Application of industrial automation techniques to precision agriculture
  - Potential to improve profitability [1] and sustainability [2] of modern farming
  - Use of PLCs (Programmable Logic Controllers), which are used to control electro-mechanical processes in industrial settings
  - Combines mechanical engineering and computer science
- The University, in contribution to this field of research, purchased and assembled two automated farming robots (FarmBots)
  - One has been successfully autonomously growing crops indoors using a temporary lighting setup (Figure 1)
- This project aims to replace the temporary lighting setup with a lighting gantry robot (Figure 3)
  - PLC-controlled servo motors on pulleys
  - Six LED (light-emitting diode) panels with variable heights
  - Ability to “grow” with plants as they mature
  - Avoid collisions with main FarmBot gantry
- A Python control module will be built to manage communication (Figure 5)
  - Control farm gantry with API (application programming interface) requests to FarmBot cloud server
  - Communicate with PLC using Modbus TCP

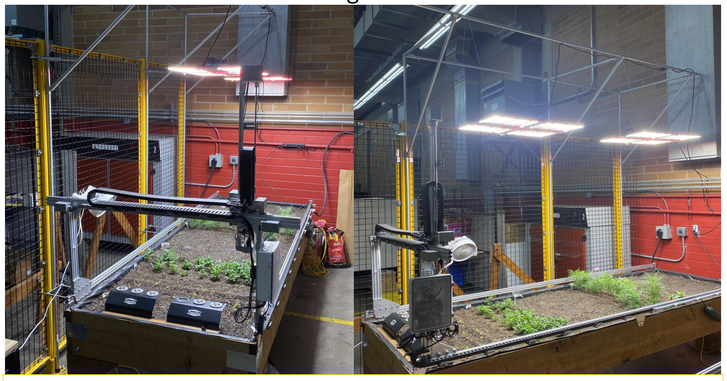

Figure 1: Operational FarmBot with temporary lighting setup

- FarmBot operates using a web-based application hosted by the parent company (Figure 2)
  - Navigates raised bed using cartesian coordinate system
  - Has camera and interchangeable heads for specific applications
  - Custom regimens for watering, weeding, planting, etc.

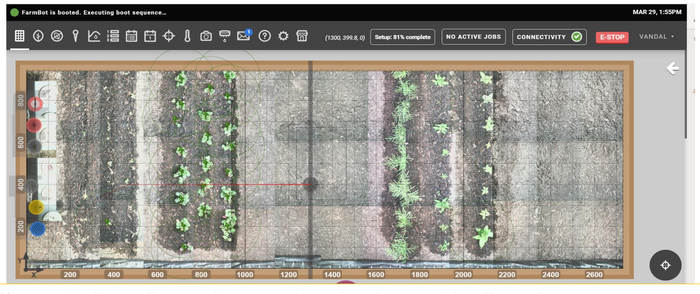

Figure 2: FarmBot online application hosted by FarmBot

Process

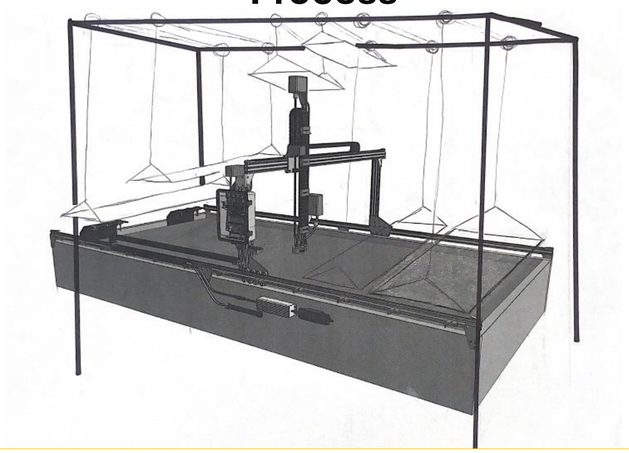

Figure 3: Proposed sketch of the new PLC-controlled lighting gantry

- API requests to communicate with FarmBot brain
  - Postman was initially used to manually receive an authentication token to access our machine (Figure 3)
  - Used to control FarmBot gantry with Python API wrapper

Figure 4: Sample API authentication token on Postman

- Modbus TCP to communicate with PLC
  - Read limit switch values which indicate if lighting gantry is up
  - Verifying limit switch values acts as a safety net to avoid collisions
  - Read height of each panel
  - Control the height of each panel using servo motors on pulleys
- Python Control Module (Figure 4) will manage both explained necessary communication

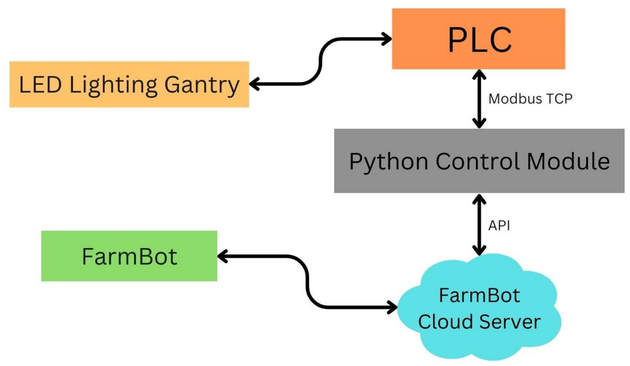

Figure 5: Python Control Module communication structure

Current Status

- FarmBot can be moved using Python Control Module
- PLC controls are outlined and will be implemented once lighting gantry is installed (Figure 6)
- The new lighting gantry robot will be installed on a second, unused FarmBot raised bed, communicating with the operational FarmBot brain to allow for safe testing
- Installation of the Lighting Gantry Robot is scheduled for May 2nd

```python
from farmbot import Farmbot, FarmbotToken
EMAIL = "vandalfarmbot@outlook.com"
PASSWORD = "AutomatedAg22"
raw_token = FarmbotToken.download_token(EMAIL, PASSWORD, "https://my.farm.bot") # Authorization token
fb = Farmbot(raw_token)                                                 # Construct FarmBot instance
class TestHandler:
    def on_connect(self, bot, mqtt_client):                             # RPCs (Remote Procedure Calls)
        LOC_X = 600 ; LOC_Y = 400 ; LOC_Z = 0
        move_request_id = bot.move_absolute(LOC_X,LOC_Y,LOC_Z)                  # Move to location
        print("*  Move to location request ID: " + move_request_id)
        # Modbus TCP PLC checks and ontrols will go here
        message_request_id = bot.send_message("Hello from Python Control Module!")
        print("*      Send message request ID: " + message_request_id)
        home_request_id = bot.find_home()                                       # Return home
        print("*         Move home request ID: " + home_request_id)
    def on_change(self, bot, state):                                    # When something changes
        print("                (%.2f, %.2f, %.2f)" % bot.position())
    def on_log(self, bot, log):                                         # Messages from FarmBot
        print("New message from FarmBot: " + log['message'])
    def on_response(self, bot, response):                               # IF RPC response succeeds
        print("ID of successful request: " + response.id)
        print("Current position: (%.2f, %.2f, %.2f)" % bot.position()+"\n")
    def on_error(self, bot, response):                                  # If RPC request fails
        print("ID of failed request: " + response.id)                           # Check which RPC failed
        print("Reason(s) for failure: " + str(response.errors))                 # Error messages
handler = TestHandler()
fb.connect(handler)                                                     # Connect to FarmBot with new handler
```

Figure 6: Sample Python Control Module Program

Future Research

- Set up private local server for FarmBot to run on
- Duplication of the gantry robot over a second FarmBot, which is currently being assembled

References

[1] M. Boehlje. (2021, February 22) The Value of Data/Information and the Payoff of Precision Farming. [Online]. Available: https://ag.purdue.edu/commercialag/home/resource/2021/02/the-value-of-data-information-and-the-payoff-of-precision-farming/

[2] P. A. C. T. Force. (2021, November 10) Task Force for Reviewing the Connectivity and Technology Needs of Precision Agriculture in the United States.” Available: https://www.fcc.gov/sites/default/files/precision-ag-report-11102021.pdf

Acknowledgements

Thank you to the Mechanical Engineering students who worked to design and build the lighting gantry: Robert Carne, Senami Hodonu, and Kathryn Reece. Thank you to Garrett Wells, a CS MS for his help researching the API request aspect of the communication structure. The original FarmBot was purchased by a donation made by John Stone to the Coeur d'Alene Robotics Fund, and the latter lighting gantry research and assembly was funded by an OUR award Spring 2023.

Learn more about me, my research project, and keep up with my future projects here: