ASHESI UNIVERSITY

COMPUTER AIDED HYDROPONICS WITH REINFORCEMENT LEARNING

UDERGRADUATE CAPSTONE

B.Sc. Computer Engineering

Stephan Nyarko Ofosuhene

2019

ASHESI UNIVERSITY

COMPUTER AIDED HYDROPONICS WITH REINFORCEMENT LEARNING

CAPSTONE PROJECT

Capstone Project submitted to the Department of Engineering, Ashesi University College in partial fulfilment of the requirements for the award of Bachelor of Science degree in Computer Engineering.

Stephan Nyarko Ofosuhene

2019

DECLARATION

I hereby declare that this capstone is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

……………………………………………………………………………………………

Candidate's Name:

……………………………………………………………………………………………

Date:

……………………………………………………………………………………………

I hereby declare that preparation and presentation of this capstone were supervised in accordance with the guidelines on supervision of capstone laid down by Ashesi University College.

Supervisor's Signature:

……………………………………………………………………………………………

Supervisor's Name:

……………………………………………………………………………………………

Date:

……………………………………………………………………………………………

Acknowledgements

I would like to acknowledge my supervisor Dr Kenobi Morris for the guidance he gave me throughout this project.

This project made use of modified code by Dr Ayorkor Korsah which was provided as part of a lecture on reinforcement learning.

Abstract

Hydroponics, a system of growing crops without soil, has been successfully used to grow crops on a commercial scale. Hydroponics has the potential to fill the gap of low agricultural production in Ghana due to its high efficiency while serving as an environmentally friendly alternative to soil culture. This method of farming has benefitted from new technologies like IoT and machine learning that make it possible to integrate intelligent agents in the management of hydroponic systems as well as collecting live data. These technologies allow for increased automation and refined control of hydroponic systems. This paper details the development and implementation of a hydroponic system equipped with intelligent agents for control and internet-enabled monitoring, data collection and storage.

Table of Contents

Table of Figures

1 Chapter 1: Introduction

1.1 Background

1.1.1 Hydroponics

Hydroponics is a method of growing crops without soil in a controlled environment [1]. This system uses nutrient solutions to supply plants with their nutrient requirements while supporting the crops in inert media [1]. Hydroponics is typically practised in greenhouses which minimises exposure to pests and is, therefore, capable of eliminating the use of pesticides [2].

1.1.2 Hydroponics vs Soil Culture

Hydroponically grown crops tend to produce more yield and use less water than their counterparts cultivated in the soil [3]. This difference is because hydroponic crops are supplied with their nutrients directly through a nutrient solution, meaning they do not have to grow large roots to have access nutrients as they do in soil culture [4]. This means they can use most of their energy on to develop the parts that appear above the soil which includes leaves and fruits. Hydroponic culture also allows the use of smaller crop spacing [3] and can be expanded vertically [5], thereby, significantly reducing the amount of land required to grow crops. Finally, less water is required for growing crops in hydroponic systems as compared to growing in the soil [3] because water is continuously cycled in the system instead of being allowed to seep into the soil and evaporate. [4]

Furthermore, hydroponics is more environmentally friendly than growing crops in soil because it uses fewer resources and crop yield can be increased without necessarily increasing the amount of land used thereby reducing deforestation [4]. It also requires less labour because activities like weeding and application of pesticides are eliminated.

Despite the overwhelming benefits of hydroponics, the initial cost of setting it up is high and requires specialised skills. This cost mostly comprises of specialized equipment for cultivation and, in this case, IoT devices and other computing resources for intelligent control and monitoring. The long term benefits of hydroponics, however, tend to outweigh the cost especially because it is capable of producing good yield all year round [5].

1.1.3 Internet of Things (IoT)

IoT is a term that refers to a class of internet-enabled things that are not traditional computers like laptops and mobile phones [6]. These devices such as fridges, light bulbs and door locks are everyday things that, with the help of an internet connection, can be automated or controlled remotely. Apart from automation and remote control, IoT is used to collect data with the use of sensors. The ability to autonomously and reliably collect data with IoT devices make them ideal counterparts for data-intensive applications like machine learning and big data.

IoT devices have been used to collect data for hydroponics using DIY systems [7] as well as in MIT's OpenAg initiative [8] to automate data collection. Some conventional processes can also be automated using IoT devices. The technology also introduces the ability to monitor and control things remotely. This reduces the need for physical interactions with the system and makes it easy to detect unexpected malfunctions as and when they occur.

1.1.4 Machine Learning

Machine Learning (ML) is a computational method used for modelling the underlying patterns in data. This is done by training a model with either labelled or unlabeled data to enable it to make predictions about previously unseen data [9]. This method of extracting meaning out of data tends to be more accurate than standard statistical methods and has higher accuracy and is faster than humans. This makes it ideal in scenarios where large data sets need to be analysed to extract meaning [9].

ML algorithms are usually supervised or unsupervised [9]. Supervised algorithms are tasked with identifying a suitable label for some labelled data. Unsupervised algorithms, on the other hand, are trained on data with no labels and are tasked with finding patterns in the data usually by clustering similar data points. They can also be tasked with identifying which cluster a previously unseen data point may belong to. Some ML algorithms like linear regression can be used to train on continuous data [9].

1.1.5 Self-Optimizing with Reinforcement Learning (RL)

The cultivation of crops in a hydroponic system requires that some conditions are maintained at some desired level. The conditions under which crops are grown as well as the constituents and the proportion of the nutrient solution affect the yield of the crops [5]. Self-optimization in this project refers to the use of RL to find effective ways of controlling a hydroponic system's system variables to optimize a goal. RL refers to a class of machine learning algorithms that learn from experience by interacting with an environment and gaining rewards when they achieve a goal [10].

1.2 Problem Statement

Over the years, Ghana has imported large amounts of food products to satisfy demand. Some of this has been attributed to taste for foreign goods as well as low productivity in the country's agricultural sector [11]. According to data collected on imports, Ghana imports 56bn CFA [12] worth of tomatoes from Burkina Faso. That is, over GHS 495.8m is spent on the import of tomatoes. This level of import is undesirable because the Ghanaian economy is heavily dependent on imports [13]. Given the amount of arable land available in the country, it should be possible to slash the number of imports significantly if production increased. Such an increase in agricultural production would most likely be accompanied by an increase in farmlands, thereby, increasing deforestation. This sequence of events will be detrimental in the long run and can be mitigated with the use of hydroponic culture which not only requires less land and water than soil culture but can be expanded vertically. Vertical expansion coupled with higher crop densities will reduce deforestation and wastage of water due to lower water requirements in hydroponic systems.

1.3 Motivation

The solution to this project is to use hydroponics as a means of agricultural production given that it is inherently more efficient than soil culture. This may seem counterintuitive because Ghana already has massive amounts of arable land. It should be noted, however, that an agricultural expansion of such magnitude would require a significant amount of deforestation. This is undesirable given that illegal small-scale mining is already claiming large portions of the nation's natural flora. The issue of deforestation can be mitigated with hydroponics because the system allows for vertical farming as well as high planting density. Hydroponics is also more environmentally friendly because it uses less water than soil culture and can be powered with clean sources of energy thereby reducing its carbon footprint.

1.4 Scope

This project seeks to implement a prototype hydroponic system with RL for control of system variables to serve as a proof of concept. This would include setting up the hydroponic system to support crop growth as well as the integration of the IoT system to collect data and automate the control of the solution pH. The implementation of a cloud backend for remote monitoring and control also falls within the scope.

The system will be fitted with sensors to monitor the pH, flow rates, ambient humidity, ambient temperature and solution temperature. These measurements will be used to control pumps that control the flow of acids and bases into the nutrient solution for pH control based on the output of the RL model.

2 Chapter 2: Design

2.1 Design Objective

The goal of this design is to create an IoT enabled hydroponic system that is capable of controlling its system variables like pH and temperature without, or with minimal, human intervention. The system is also expected to collect data throughout crop growth. The implementation of the system should;

- Be capable of automated real-time data collection
- Be able to send collected data to a cloud server for storage
- Have a dashboard that pulls live data from the server for remote monitoring over the internet
- Have three hydroponic subsystems that can be supplied with different nutrient solutions from different reservoirs
- Each subsystem should be able to circulate the nutrient solution to support crop growth
- Have an intelligent RL agent that is capable of automated pH control

The use of three subsystems in the same hydroponic system makes it possible to grow the same crop with different nutrient solutions. This would increase the variety of data collected per growing season about how crops fare when grown in different solutions.

2.2 Materials

The primary material for the hydroponic setup is PVC pipes because they are easily accessible and workable. The supporting frame was made of ¾-inch (diameter) pipes, and three 4-inch pipes were mounted on the frame to serve as the subsystems (see Figure 2 for an image of the frame with mounted hydroponic subsystems). Unlike PVC pipes, wood and metal require an extra finish to protect them from weather conditions. They also need special techniques for joining such as welding or nailing whereas PVC pipes can be joined with glue using joints (see Figure 5 for an image of the joints used to join the pipes).

The nutrient solution would be circulated with 12V self-priming pumps. Self-priming pumps can pump fluid regardless of their relative height, thereby allowing some flexibility in design decisions. They are also reliable in situations where fluid levels are low.

Some inert materials have been used to provide support for crops in hydroponic systems such as perlite and gravel [4]. Gravel was chosen for this project instead of perlite because it is cheap and easy to obtain.

2.3 Design Decisions

2.3.1 IoT

The IoT system for this project will consist of Arduino microcontrollers (Arduino Uno) and a raspberry pi with each device connected to the raspberry pi via a Bluetooth network. The IoT system will be connected to a server and will send data collected to the said cloud. A Bluetooth network was chosen because of the low energy requirement and the fact that all devices will be nearby.

The two microcontrollers (Arduino and raspberry pi) will both serve unique purposes in the system. The Arduinos will be responsible for interfacing with the sensors in the system and controlling actuators. This is because these devices have analogue to digital (ADC) converters and digital peripherals. The raspberry pi, on the other hand, will be the mother computer which will aggregate the data collected by each device (via Bluetooth) and transmit them to a server on the cloud. This function is assigned to the raspberry pi because it has an operating system that allows high-level programming and also has built-in hardware for internet communication using Wi-Fi and LAN.

The server is responsible for storing data from the IoT system and serving them up to a user for live monitoring. The server also serves as an intermediary between the user and the IoT system to enable remote control. Remote monitoring and control will be done via a dashboard provided by the server. The server may also be responsible for handling the machine learning workload if it proves to be too much for the raspberry pi. The image below shows a graphical representation of the IoT system.

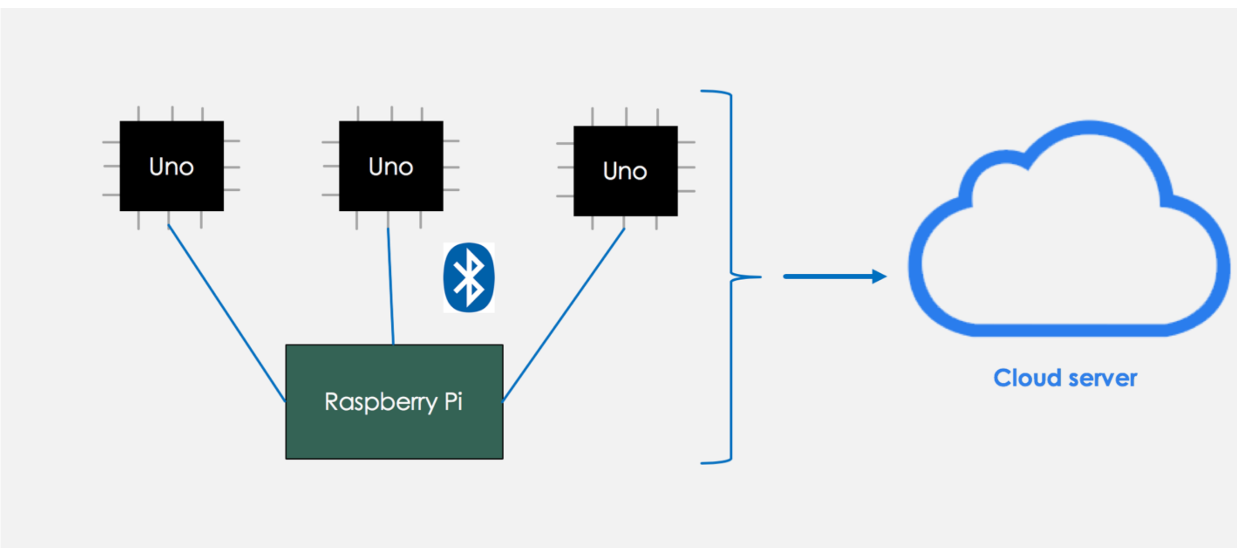

Figure 1. Design of IoT System

2.3.2 Hydroponic System

The hydroponic system will be based on the Nutrient Film Technique (NFT) which provides support for crops with an inert material and circulates nutrient solution for the crops to absorb. The crops, with their supporting material, will be mounted in 4-inch pipes connected to a reservoir, and the circulation will be done with the use of an electric pump.

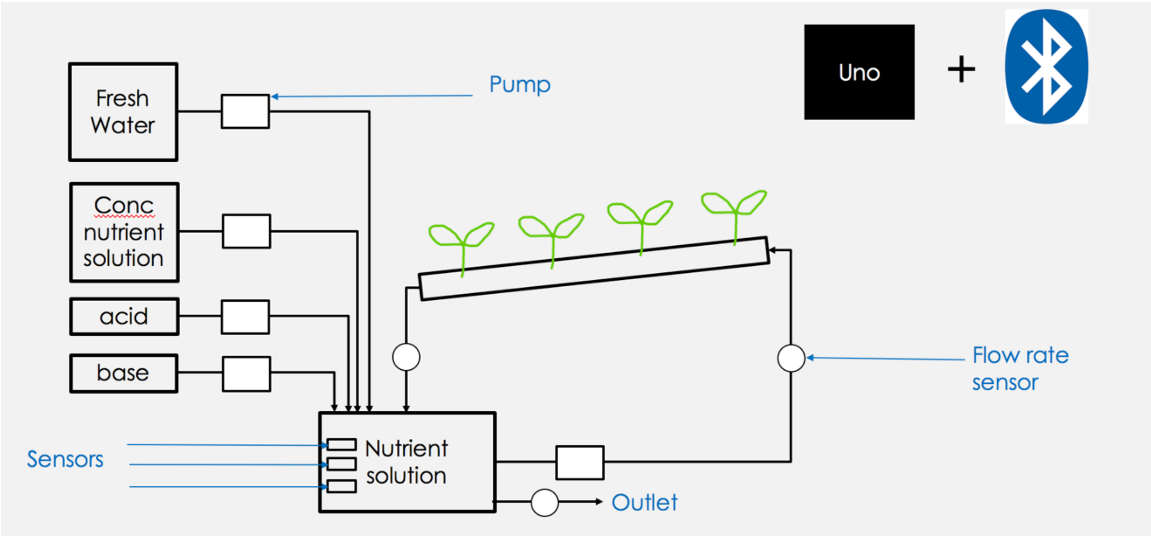

Figure 2. Design of the Hydroponic System

The pipe will have flow rate sensors at both ends to monitor the rate of inflow and outflow of the solution. This will be used to control the rate at which the solution is pumped into it considering that growth of roots will slow down the passage of solution over time. The rate at which the solution flows into the pipe should be equal to the rate at which it flows out of the pipe to prevent it from overflowing. Data collected from these sensors will also be useful for detecting malfunctions like blockages or leakages in the system.

2.3.3 pH Control with Reinforcement Learning

The intelligent agent for this project was developed with Q-learning. The Q-learning algorithm allows an agent to explore and learn from an environment and gain experience [14] by filling a Q-table. The dimensions of the Q-table are based on the number of possible actions and states in the environment. Each cell in the Q-table contains a Q-value which shows the long-term reward of taking an action in a certain state. The values in the Q-table encode the learned policy of the agent, which is, the rules by which the agent reaches its goal [14]. This policy is learned by obtaining positive and negative rewards in the environment after a goal is reached or an action is performed.

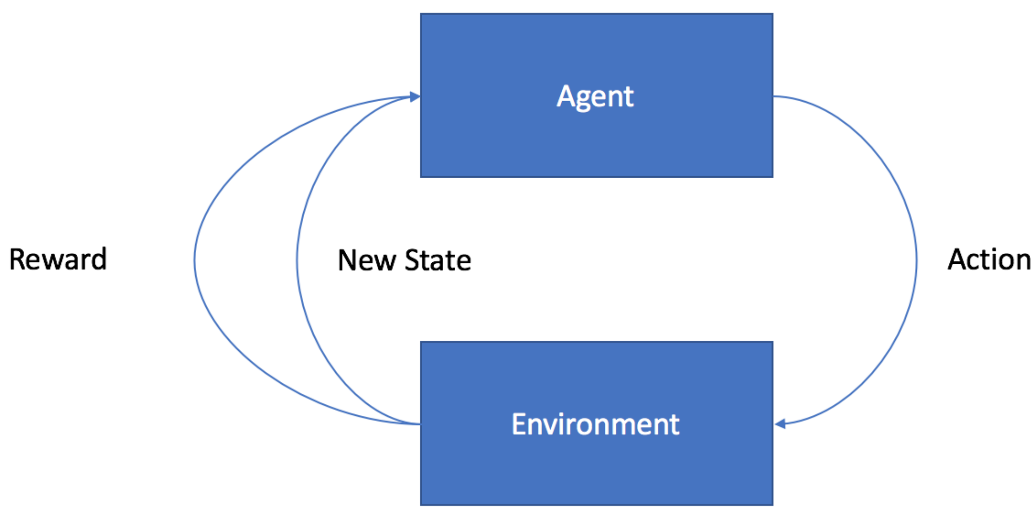

Figure 2. Markov decision process for reinforcement learning

At this point, you may have realized that it would be difficult to store the continuous states of a solution's pH in a discrete table. To address this problem, the pH range is divided equally into several quantized levels to make continuous pH values discrete. The larger the number of divisions, the better the approximation of the pH value.

Using an actual Q-table to store the state action mappings would require a lot of space if the environment has a large number of possible states. Both a neural network and a Q-table were used in training the Q-learning algorithm. The outcome of both approaches can be found in the results section.

Because of the unexpected behaviour of intelligent agents, while learning, the agents were trained in a simulated environment. To use such intelligent agents in a live system for controlling pH, the simulations in this project will be used as a way of finding a suitable way of training Q-learning agents before deploying them in live environments. The approach used in this project will test the hypothesis that the experience of an agent trained in one environment can be transferred to a similar environment. This transfer of experience should be possible the policy of similar environments should be similar. The expectation is that transferring the experience of agents should reduce the number of epochs required to reach mastery. This would be a good step toward making RL agents feasible to use in a real-world environment.

The agent would get a reward of 100 if it reached the desired goal state. Also, it gets a reward of one if its action moved it closer to the goal state and a penalty of -5 if it moved away from the goal state. Furthermore, the agent incurs a penalty of -500 if it exceeds the maximum or minimum bounds of the environment. Finally, the agent is capable of three actions, move up with a step of one or move down with a step of one. An epoch ends if the agent reaches the goal state or exceeds one of the boundaries.

Two environments were created for this purpose; A linear environment and a quadratic environment. The linear environment can be thought of as a liquid reservoir with an inlet and an outlet. The goal of the agent is to get the level of the liquid from a random point to the desired range by draining the liquid through the outlet or supplying more liquid through the inlet. The quadratic environment is similar to the reservoir analogy; however, this environment has a quadratic response. To test the hypothesis, an agent will be trained on the linear environment, until it achieves a satisfactory level of mastery. Since this agent has no previous experience with any previous environment, it is expected to require a large number of epochs to learn how to achieve its goal. After the agent has been trained on the linear agent, its Q-table will be used to train on a quadratic environment. Since the agent trained in the quadratic environment has previous experience with a similar environment, it will be expected to reach mastery in a significantly smaller number of epochs. Refer to the results section for the outcome of this proposed approach to training RL models.

3 Chapter 3: Implementation

3.1 Overview

This chapter provides an overview of the parts of the initial design that were implemented in the prototype. It will also break down the details of how each component was implemented and the reasons behind the implementation decisions.

3.2 Prototype

3.2.1 Hydroponic System

As mentioned in the previous chapter, the hydroponic system was implemented with PVC pipes. The hydroponic system consists of a supporting frame, which was made of ¾ inch pipes, and 4-inch pipes were used to support the crops and carry the nutrient solution. The dimensions of the frame were 1.5m x 1m x 1m and were joined with only T joints and elbow joints.

Figure 3. 3/4-inch T elbow and T joints used to build the frame for the hydroponic system

The length of the 4-inch pipe was 1.5m and was sealed at one end with a 4-inch end cap, and the other end was fitted with a combination of reducers that reduced the diameter of the pipe from 4-inches to 1-inch. The combination of reducers used was, 4-inch by 2-inch reducers, 2-inch by 1-inch reducers and 1-inch by ¾-inch reducers. The reducers were joined to each other with pipes of the corresponding dimension. That is, the 4-inch by 2-inch reducer and the 2-inch by 1-inch reducers were joined together by inserting a glueing a 2-inch pipe in the corresponding side. This had to be done because the reducers were not designed to connect directly.

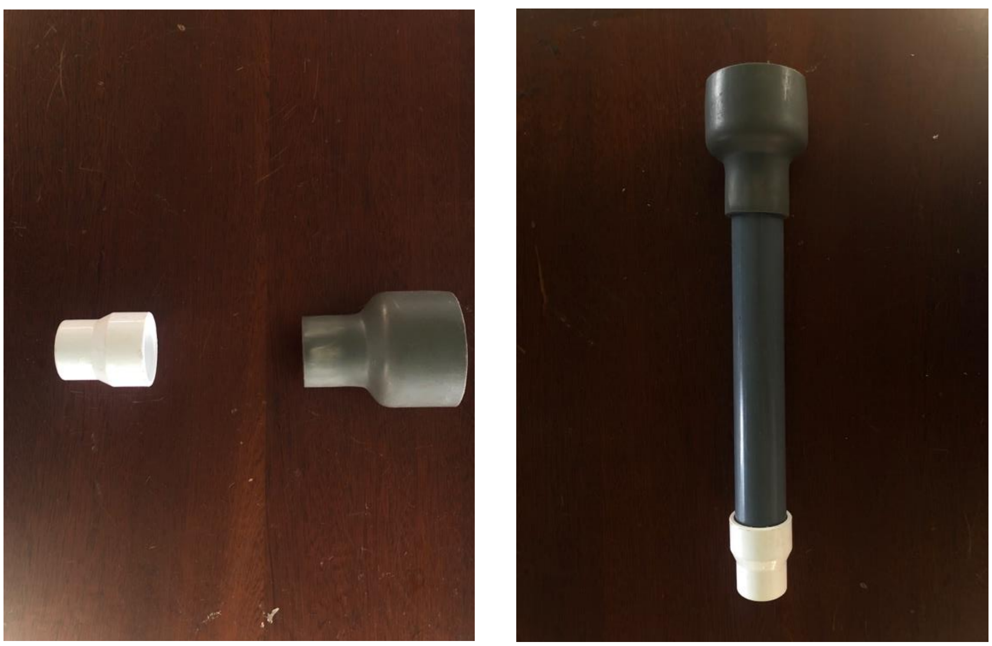

Figure 4. Reducers before and after being connected with a 1-inch pipe

This process was repeated until the 4-inch pipe was reduced to ¾-inches as seen in the image below. The pipes connecting the reducers were cut to fit the reducers which prevented them from showing. A ¾-inch elbow joint was then fitted at the end of the reducer combination to direct the flow of liquid into the reservoir.

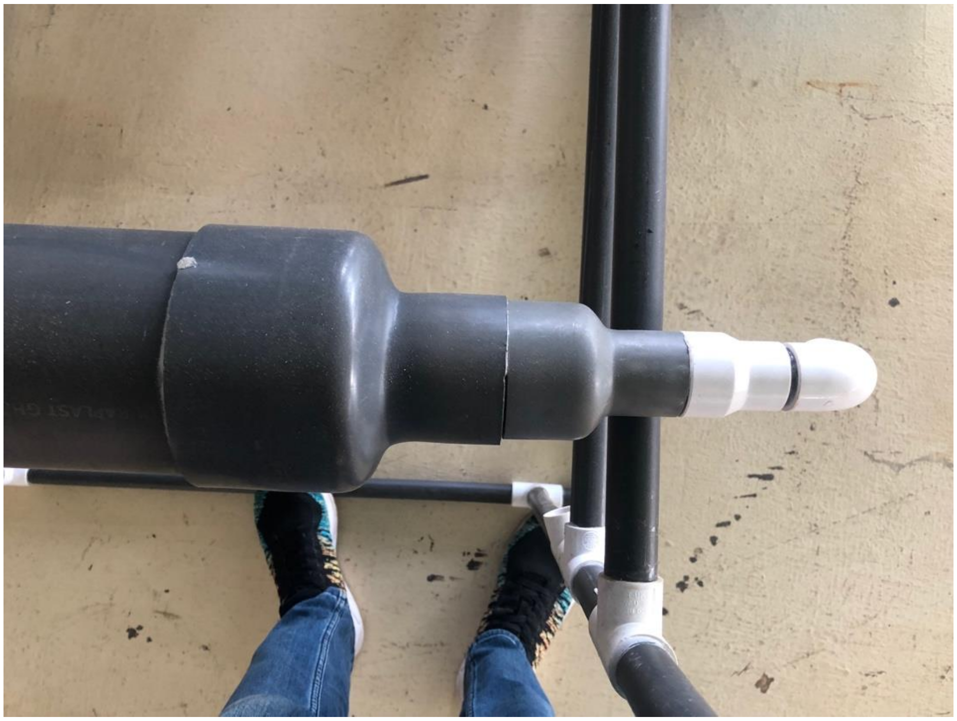

Figure 5. Reducing the 4-inch pipe to 3/4-inches

The 4-inch pie is inclined with a 10% gradient. The circulation of the nutrient solution is achieved by pumping the solution with a self-priming pump at a rate of 1 gallon per minute from the reservoir into the end of the pipe sealed with the end cap (the higher end). The solution then flows downward and out of the end with the reducers back into the reservoir through a ¾-inch pipe. The 4-inch pipes also have 15cm diameter holes cut out in them to serve as a way of mounting crops and their inert media into the circulating nutrient solution. The holes were cut to allow a crop spacing of 20cm. Finally, the 4-inch pipes are fastened to the frame at the high end with braces.

Figure 6. Image showing inlet to the pipe, the fastening braces and the cut-out holes for crops

3.2.2 IoT System

The IoT system is made up of three main components; the microcontrollers, that interface directly with the hydroponic system, the raspberry pi that aggregates the microcontrollers using a Bluetooth connection and a cloud server that is responsible for receiving and storing data sent from the raspberry pi. The cloud server also enables remote monitoring over the internet using the dashboard which will be discussed in the next section.

The microcontrollers in the IoT system are mainly responsible for collecting data from the connected sensors and performing control instructions received from the raspberry pi over their Bluetooth connection. The data collection from the sensors is performed by sampling the sensor values once every second. The raspberry pi collects data from the microcontrollers by sending a request for data using a round robin algorithm. Once a request is received, the microcontroller responds with the last recorded sensor readings as a JSON string. If more than one request is received within the same second, the microcontroller will respond with a message indicating that no new data is available. The Arduino Uno microcontroller was used for this prototype.

The microcontroller also executes control instructions received from the raspberry pi. This is done by parsing and decoding the messages received and executing the instruction. For example, the microcontroller may receive the message “/control?acid=5” which is interpreted as "turn on the pump responsible for supplying acid into the reservoir for 5 seconds." The format of messages that are supported by the microcontrollers is detailed in the appendix.

As mentioned above, the raspberry pi is responsible for aggregating the individual microcontrollers. It collects the data from the microcontrollers via a round robin where it periodically requests for available data form the devices. It then sends the data to a cloud server for storage and remote access. The raspberry pi is also responsible for running the neural networks and sending control information to the microcontrollers based on the output of the neural networks.

The final component in the IoT system is the cloud server. The server was built with Python using a framework known as Flask which enables python to handle HTTP requests on a server. This framework was chosen because of its simplicity and the ease of adding more libraries to extend its functionality.

The server contains the models, which are database objects, that define the structure of the data to be stored. The models are defined using an Object Relational Model package known as mongoengine which allows data to be persisted a MongoDB database. MongoDB was chosen for this project because of its flexible NoSQL design which allows information in the database to be dynamic. In other words, models are easy to design because it uses a document-oriented structure, does not require a schema and all fields in a document do not have to be present for any given document. The lack of a rigid structure makes MongoDB ideal for this project because it allows for selective use of sensors without breaking the system. A list of metrics or sensor readings supported by the system can be found in the appendix.

Apart from defining the models that provide the structure of the data that can be stored in the database, the server provides Representational State Transfer (REST) Application Programming Interfaces (APIs) for sending data for storage and requesting data. The available API endpoints for the cloud server are detailed in the appendix.

3.2.3 Dashboard

The dashboard for the system is designed to allow a user to remotely monitor the behaviour of the system remotely over the internet. The dashboard simply requests data from the server using the REST APIs and displays the data as a graph. The dashboard is meant to show the most recent sensor readings from the hydroponic system by showing both a general overview of the data for all systems or system specific data. The image below shows the current state of the dashboard prototype.

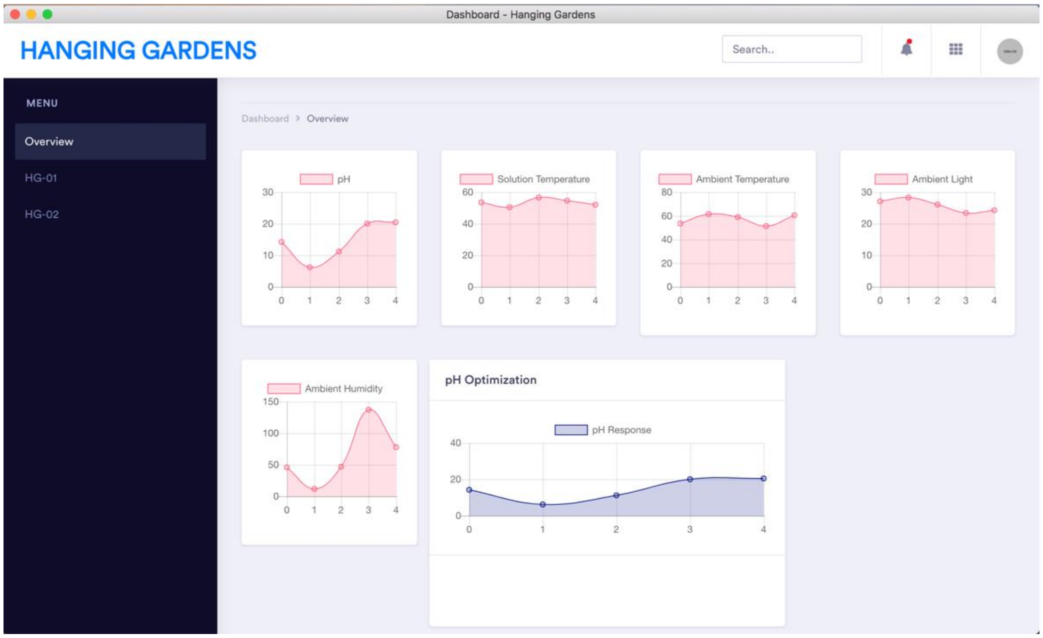

Figure 7. Dashboard Prototype

The dashboard was implemented as a desktop application using a framework known as Electron. Electron allows the development of cross-platform desktop applications using web technologies. That is, the frontend of is written in HTML and CSS, and the backend is written in JavaScript. This framework is convenient for application development because it allows the use of the well-developed web programming ecosystem for local app development. This reduces the effort required to build an application because the developer is not required to learn a new User Interface (UI) package.

4 Chapter 4: Results

4.1 Dashboard

The prototype dashboard for the system is capable of collecting data (sensor readings of system variables) from the cloud backend and displaying it in graphs. As can be seen in Figure 9, the dashboard shows an overview of all the data available in the system. This overview is an average of sensor readings from all available systems which allows a user to have a high-level view of the behaviour of available systems. There is also the option of viewing system variables of specific systems. This can be accessed by clicking the menu item corresponding to the desired system. Currently, the dashboard prototype only requests for data when the desired system name or the overview is clicked. However, it should be modified to show live data as it arrives on the server if it should be used in a live system.

4.2 IoT System

The IoT system is divided into two sections, the Arduino microcontrollers that interface directly with the hydroponic system and the raspberry pi that aggregates the Arduinos and relays information to the server. The delay in communication in the round robin is reduced by segmenting code based on time. That is, after the sensor readings are taken by the microcontroller, it remains idle and waits for the next instruction. The microcontroller is also able to avoid resending old data by checking whether or not the most recently available data has already been sent. This ensures that only fresh and live data is transmitted from the microcontroller.

4.3 Reinforcement Learning

Using RL in the real world live environments pose some challenges such as random behaviour of the model before it learns a reasonable policy. Also, continuous valued states pose a challenge in training RL models because they cannot explore an infinite state space. The approach used in this project attempts to address these issues.

The RL model was initially trained using continuous valued states and a neural network as a function approximator. Using this approach required a relatively large number of epochs to converge at a useful policy. Also, the previous actions of the model had to be stored and used to retrain the neural network after each epoch to enable it to learn from its experience. This was slow and inefficient and increased the amount of time required for each epoch. For this reason, neural networks were replaced with traditional Q-tables because they provided faster turnaround times for running experiments.

The transfer learning experiment, as explained in the design section, was performed by training an agent on a linear environment after which the Q-table obtained from the training was used to train on a quadratic environment. In other words, the experience the agent obtained from interacting with the linear environment was transferred to the quadratic environment. The agent trained in the linear environment took 37000 epochs to reach mastery. The probability of reaching the goal state after training on the linear environment was 0.998. The learning curve and the goal curve are shown in the figures below. The learning curve is obtained from averaging the maximum Q-values while the goal cure was obtained by calculating the ratio of successful tries to unsuccessful tries after each epoch.

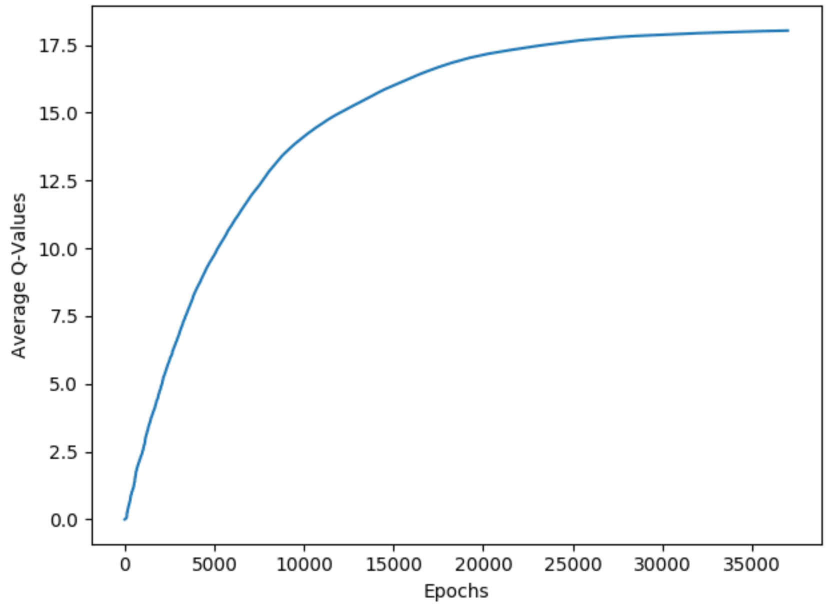

Figure 8. Learning curve for inexperienced agent trained in a linear environment for 3700 epochs

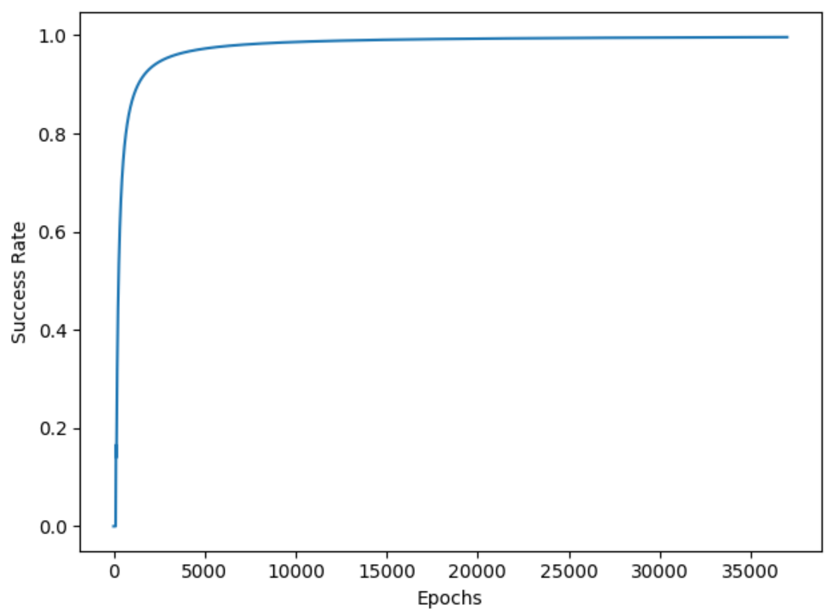

Figure 9. Goal curve for agent trained in the linear environment for 37000 epochs

The graphs below show the goal curve and the learning curve of one transfer learning agent. From the graph of the learning curve, it appears that the agent is unlearning the policy it learned from the previous environment. The pattern of unlearning in the transfer model may be due to the costs it incurs by using the "assumptions" that worked in the old environment on the new one.

Also, it appears to be learning from the new environment simultaneously as shown by the goal curve.

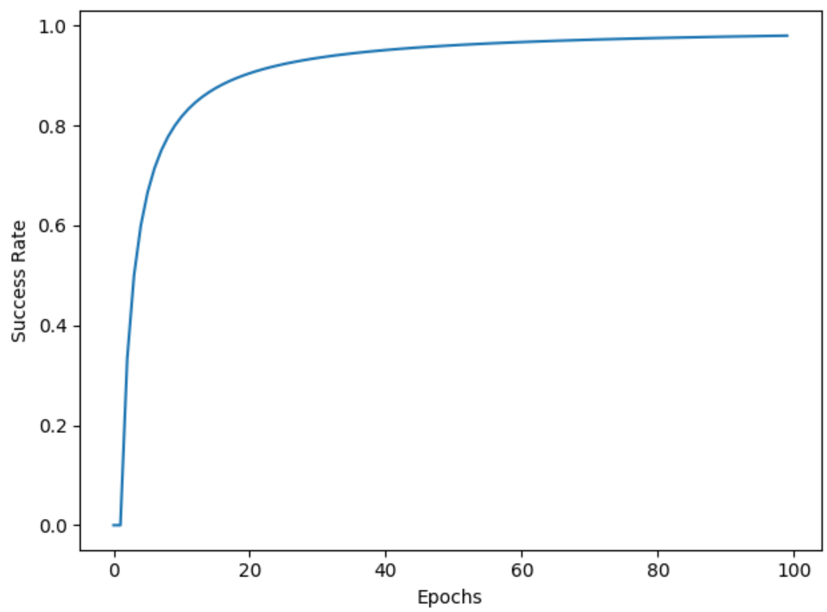

Figure 10. Goal curve for transfer learning agent trained for 100 epochs

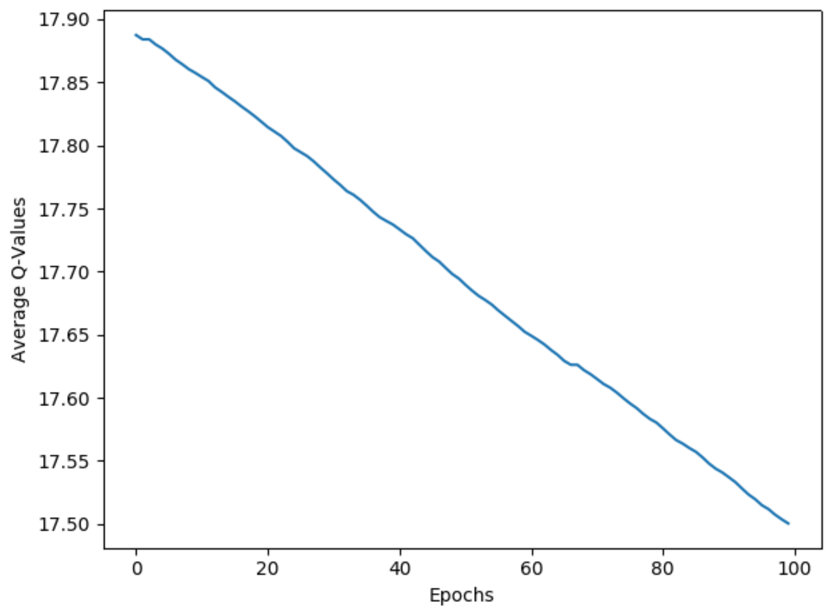

Figure 11. Learning curve for transfer learning agent trained for 100 epochs

The rationale behind transferring the experience of a model trained on a similar environment is to reduce the time required to reach mastery as mentioned in the design section. The transferred model was, therefore, trained for only 100 epochs which are relatively small as compared to the 37000 epochs used for the linear model. To compare the difference between the model performance when trained with and without previous experience, the model was without experience, also for 100 epochs. Thirty models each were trained for both agents with transferred experience and agents with no experience, and their performance was compared. The average goal rate for agents with no experience was 0.578 while the average goal rate for agents with experience was 0.808. The histograms below show that more than of the agents with previous experience goal rates that were greater than or equal to the average while agents without experience had more randomly distributed experience.

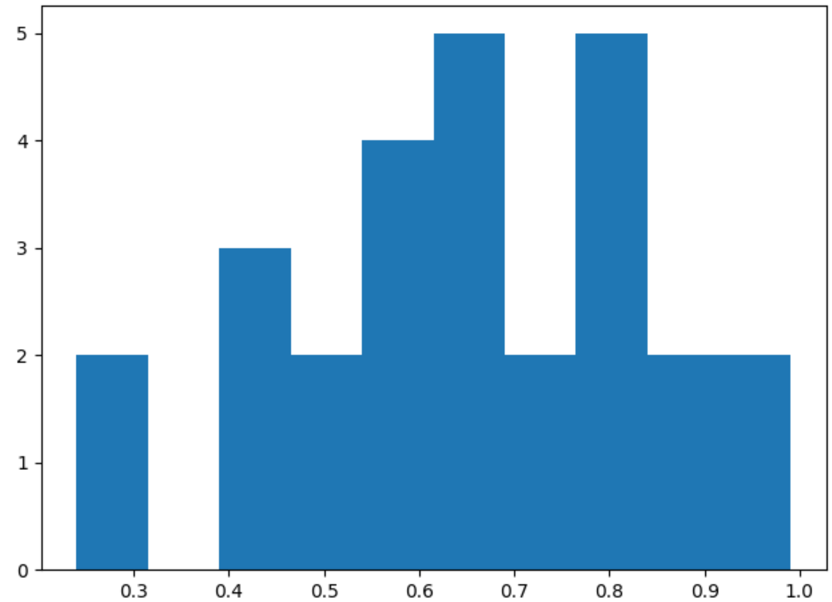

Figure 12. Histogram showing performance of agents trained without experience in the linear environment

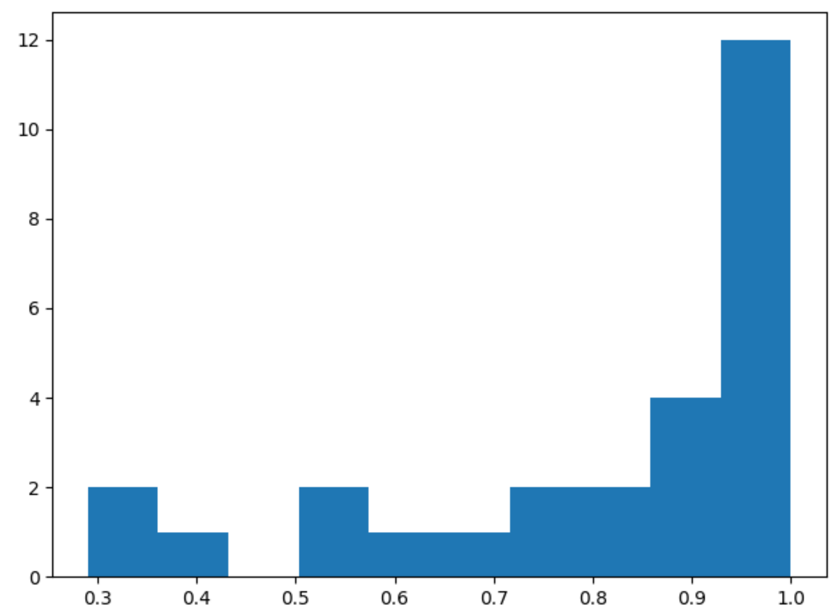

Figure 13. Histogram showing performance of agents trained with experience in the linear environment

4.4 Limitations

This project focused on building a prototype as proof of concept and tested some the hypothesis of transferring experience from similar environments in RL. As such, this project is not production ready because some components have not been fully implemented. Specifically, the dashboard does not have the feature of getting live data from the server, and no RL model was trained on the real environment. Finally, some sensors were not implemented in the IoT system even though the server has been programmed to handle data from them.

4.5 Discussion of Similar Works

Due to the difficulty in developing classical models for pH control, several alternative methods have been developed to address this challenge. Methods such as fuzzy logic with neural networks and reinforcement learning have been used for pH control.

4.5.1 Fuzzy Logic and Neural Networks

Control of pH and Electrical Conductivity (EC) has been achieved only with fuzzy logic [15]. The rules for controlling pH was encoding in 12 fuzzy rules which led to suitable results [15]. This method required previous knowledge of how solution pH is controlled by an expert and inexperienced farmer. This approach requires the use of data that may not be available and limits the system's performance to what is possible by human standards and human learning.

A similar approach to pH control used both fuzzy logic and neural networks [16]. This method also required data on how experts controlled solution pH. This in turn limited the ability of the system to control pH to human learning.

Despite the shortcomings of the previously mentioned approaches, they were able to smoothen the changes in pH.

4.5.2 pH Control with Reinforcement Learning

A similar approach to pH control with reinforcement learning has been implemented using softmax and epsilon-greedy policies [17]. Unlike the approach used in this project, the reward of the agent was calculated based on how far it was from the goal state. The environment also had fewer states than what was used in this project (5 states compared to 140 states used in this project). Finally, the agent was capable of more actions (12 actions compared to the two actions in this project).

Despite the differences, this approach also yielded good results in simulation given that it was able to reliably reach the goal state [17]. The approach discussed was not used in a live environment and did not have any considerations for reducing the amount of time required for the agent to learn a reliable policy.

5 Chapter 5: Conclusion

The goal of this project was to develop a prototype for a hydroponic system as a way of providing an alternative means of growing crops. The motivation for developing this system is the low levels of agricultural productivity in Ghana and the environmental impact of soil culture. This project has detailed the design and implementation of a prototype hydroponic system as well as a method of reducing the time required to train an RL agent in a live environment.

The prototype has fulfilled most of its design objectives such as automated data collection which is performed by the IoT system and data storage in a cloud server. The dashboard for viewing data collected for system variables was also implemented. However, it is currently not equipped for showing live data. And finally, the hydroponic system has three subsystems for growing crops with different nutrient solutions.

5.1 Future Works

Despite the achievements of this project, there is still a lot to be done to improve the system and make it production ready. To achieve this, the system should achieve the following;

- Be able to show live data on the dashboard
- The dashboard should provide notifications of malfunctions like leakages and whether the system is online
- The IoT system should support all sensors implemented on the server. See the appendix for a list of sensors the server can handle
- An RL model should be implemented for controlling pH
- The system should be capable of climate control

- An RL model should be created to help improve the yield of crops
- Arduino should be replaced with more powerful microcontrollers with industry grade ADCs

6 APPENDIX

6.1 Web API Specifications

This section describes the APIs for interacting with cloud server. The server works with HTTP requests.

<u>Get data from the server using paging</u>

API endpoint: /api/data

Request method: GET

Response type: JSON

Parameters:

page_number: int

page_size: int

Special values for page

$all:

Gets data ordered by time received in the server.

$end:

Gets data ordered by time received on the server. Separate JSON documents are constructed for each available system.

$overview:

Similar to $end. However, values are averaged.

<u>Send data to server</u>

API endpoint: /api/data

Request method: POST

Response type: JSON

Accepts: JSON

Create new hydroponic systems on server

API endpoint: /api/systems

Request method: POST

Response type: JSON

Accepts: JSON

Get all available hydroponic systems on server

API endpoint: /api/systems

Request method: POST

Response type: JSON

6.2 IoT API specifications

This section defines the API for communicating with the Arduino over Bluetooth. The raspberry pi sends a string and gets a response (string) from the microcontrollers.

Get data from microcontroller

API: /data

Response type: JSON

{system_id: string,

```
    {
    ambient_temperatue: float,
    humidity: float,
    solution_level: float,
    acid_level: float,
    base_level: float,
    solution_temperature: float,
    acid_flowrate: float,
    base_flowrate: float,
    }
}
```

Control instruction for adding acid

/control?acid=10

Turn on the acid pump for 10 seconds and turn it off. The amount of acid added will depend on the flowrate of the pump.

/control?base=10

Turn on the base pump for 10 seconds and turn it off.

References

[1] P. A. CHATELIER, “HYDROPONICS,” *Sci. Teach.*, vol. 21, no. 2, pp. 74–75, 1954.
[2] M. D. Mbaga, “The Prospects of Sustainable Desert Agriculture to Improve Food Security in Oman,” *Consilience*, no. 13, pp. 114–128, 2015.
[3] G. Barbosa *et al.*, “Comparison of Land, Water, and Energy Requirements of Lettuce Grown Using Hydroponic vs. Conventional Agricultural Methods,” *Int. J. Environ. Res. Public. Health*, vol. 12, no. 6, pp. 6879–6891, Jun. 2015.
[4] “HYDROPONICS: Crops without soil,” *Spore*, no. 150, pp. 20–20, 2010.
[5] A. Dalvi and R. Patel, “Research on Food Computer: New Step towards Technology Lead Farming,” vol. 9, no. 2, p. 2, 2018.
[6] S. Ornes, “The Internet of Things and the explosion of interconnectivity,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 113, no. 40, pp. 11059–11060, 2016.
[7] C. Peuchpanngarm, P. Srinitiworawong, W. Samerjai, and T. Sunetnanta, “DIY sensor-based automatic control mobile application for hydroponics,” in *2016 Fifth ICT International Student Project Conference (ICT-ISPC)*, Nakhon Pathom, Thailand, 2016, pp. 57–60.
[8] E. Castelló Ferrer *et al.*, “Personal Food Computer: A New Device for Controlled-Environment Agriculture,” in *Proceedings of the Future Technologies Conference (FTC) 2018*, 2019, pp. 1077–1096.
[9] E. Salinas, “Katie Malone on Machine Learning,” *IEEE Softw.*, vol. 34, no. 4, pp. 92–96, 2017.
[10] C. Szepesvári, “Algorithms for Reinforcement Learning,” *Synth. Lect. Artif. Intell. Mach. Learn.*, vol. 4, no. 1, pp. 1–103, Jan. 2010.
[11] M. A. Adombila, “Food imports cost US$2.4bn annually,” *Graphic Online*, Ghana, 17-Jul-2018.
[12] F. Owusu-Ansah, “Ghana Imports 56bn CFA of Tomatoes From Burkina Faso as Local Production Suffers,” *Th Chronicle*, Ghana, 27-May-2018.
[13] “Ghana Trade Statistics | WITS.” [Online]. Available: https://wits.worldbank.org/CountryProfile/en/GHA. [Accessed: 17-Apr-2019].
[14] Y. Wang, Q. Meng, W. Cheng, Y. Liug, Z.-M. Ma, and T.-Y. Liu, “Target Transfer Q-Learning and Its Convergence Analysis,” Sep. 2018.
[15] M. Fuangthong and P. Pramokchon, “Automatic control of electrical conductivity and PH using fuzzy logic for hydroponics system,” in *2018 International Conference on Digital Arts, Media and Technology (ICDAMT)*, 2018, pp. 65–70.
[16] T. Morimoto and Y. Hashimoto, “Application of fuzzy logic and neural network to the process control of solution ph in deep hydroponic culture,” *IFAC Proc. Vol.*, vol. 24, no. 11, pp. 147–152, Sep. 1991.
[17] S. Syafiie, F. Tadeo, and E. Martinez, “Model-free learning control of neutralization processes using reinforcement learning,” *Eng. Appl. Artif. Intell.*, vol. 20, no. 6, pp. 767–782, Sep. 2007.