A Novel System Architecture for Automated Field-Based Tent Systems for Controlled-Environment Agriculture

Dan Wagner
Department of Computer Science
Kansas State University
Manhattan, KS
danwagner@ksu.edu

Arslan Munir
Department of Computer Science
Kansas State University
Manhattan, KS
amunir@ksu.edu

Mitchell Neilsen
Department of Computer Science
Kansas State University
Manhattan, KS
neilsen@ksu.edu

***Abstract*—Experimentation within the field of agronomy relies upon maintaining a controlled operating environment to determine various environmental factors' effects upon a crop. These experiments are carried out in small growth chambers and can control limited variables such as light, temperature, and humidity. Space is a premium inside the chambers which limits the capacity for additional sensors and other equipment. Field conditions are more complex than a growth chamber, which makes it difficult to analyze the effect of factors in a more realistic scenario. In this paper, we propose a system architecture for a field-based controlled environment for agriculture and experimentation. First, the overall architecture is proposed for integrating a multitude of wired and wireless sensors, different controllers, small unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs), and actuators to assess and maintain environmental variables. Next, each component is detailed for its role and responsibilities within the system. Then, scientific applications of the system are proposed and explored before finally analyzing a case study implementation of the architecture.**

***Index Terms*—agriculture, embedded systems, IoT, cyber-physical systems, high nighttime stress**

I. Introduction

The controlled-environment agriculture (CEA) is an advanced form of agriculture based on hydroponics where crops grow inside of a controlled environment [1]. There is a growing demand for fresh produce that is high quality and organic. Climate changes make growing desirable crops difficult when out of season, regardless of consumer demand. The CEA remedies these problems by recreating the desired environment for optimal crop growth. Furthermore, agronomy experiments are typically carried out in controlled operating environments. Environmental factors, such as temperature and humidity, are controlled within growth chambers to analyze their effects on crops. These chambers, while effective, are constrained on space and cannot accommodate more sensors or equipment. The chambers are set to the specific requirements for the environment and not changed. The chambers are maintained by the control system within the chamber itself.

Current academic research on CEA focuses on individualized, complex systems for analyzing single-dimension interactions between crops and the environment as well as larger-scale smart farming systems with multiple individual embedded devices scattered throughout the farm area. The former exhibits reliable environmental controls at the cost of space and complexity. Larger smart farming systems allow complex farming areas to be examined and regulated but are difficult to manage for environmental control decisions. Although recent years have seen growing progress in CEA [2], an architecture that can enable monitoring and control of multiple environmental factors in farm-like conditions still needs to be developed.

To overcome the limitations of small growth chambers, and to analyze the effect of different environmental factors on crops, we propose an architecture for large, automated, field-based tent systems for CEA and experimentation. A multitude of sensors, both wired and wireless, for observing various variables can be installed and used within the tent. The required environmental parameters for the area are designated by the designer or user of the system. After the parameters' specifications, the system uses the readings received from sensors to engage or disengage actuators to maintain an optimal environment. Our main contributions are as follows:

- We propose an architecture for large, automated, field-based tent systems for CEA and experimentation.
- The proposed architecture is modular and scalable comprising of different subsystems responsible for sensing, analytics, decision-making, and actuation. We discuss the constituent subsystems of the proposed architecture.
- We discuss different applications of the proposed architecture, such as CEA, phenotyping, and precision agriculture.
- We prototype and analyze a simplified implementation of the architecture in a wheat heat stress experiment, and demonstrate the capability of our proposed architecture in studying the effect of different environmental conditions on crops in CEA settings.

II. Related Work

Recent research developments in CEA focus on optimizing horticultural lighting systems within greenhouses [2]. Cornell University's Greenhouse Lighting and Systems Engineering (GLASE) consortium is continuing research into light-emitting diode (LED) efficiencies and light wavelengths in greenhouses.

Their focus is creating integrated control systems for lighting, carbon dioxide, temperature, and humidity. More complex interactions, such as imaging data and aperture control, cannot be achieved easily without further research and revisions to existing systems. Higher-dimension data points, such as video/image data, are becoming more important and relevant in agricultural practices for increasing crop yield and quality through the use of drones and stationary cameras. A system architecture to manage the complex sensor systems and integrate them together is required for more informative experimentation.

Current academic research focuses upon individualized, complex systems for monitoring singular variables or interactions between the crop and environment, as well as larger-scale Internet of things (IoT) based smart farming systems with multiple individual embedded devices scattered throughout the area. Castañeda-Miranda et al. [3] designed an IoT system to accurately predict frost levels in a greenhouse, and to activate an anti-frost irrigation system to prevent damage and crop disaster. The authors utilized an artificial neural network (ANN) to predict the greenhouse temperature, which is then provided as input to a fuzzy logic system to determine the appropriate amount of anti-frost to apply to maintain the thawed environment. The proposed architecture in [3] can leverage our proposed architecture to create a larger, overarching smart farm system that could monitor and control multiple environmental parameters by integrating the appropriate controller(s), actuators, and communication protocols. It follows similar thought processes but incorporates each of the individualized subsystems into an overarching architecture with real-time predictions and analysis to act on environmental stimuli. Farooq et al. [4] surveyed the role of IoT in agriculture for smart farming. They noted that agricultural trends were individualized systems for the farm, field, and greenhouse, which reported results to a central server, typically cloud-based, for storing the information.

III. System Architecture

Figure 1 depicts our proposed system architecture for automated field-based tent systems for CEA. The proposed architecture comprises of the following subsystems or components: (i) Data Node, (ii) Sensor Controllers, (iii) Wireless Controller(s), (iv) Vehicle Controller, (v) Database Node, (vi) Analytics Node, (vii) Decision Node, and (viii) Actuators. The architecture is designed in a distributed control network style (Figure 1). Each location within the farming area becomes a subsystem of the overall architecture. Each subsystem is compartmentalized based upon its functional role and communicates in fixed ways. The system has the capabilities to automate irrigation, control temperature, monitor sunlight conditions, and other important factors related to crop yield via actuators and relays. The entire control system is part of a local area network (LAN). Different sensor modalities communicate over the internal network by sending information to their respective controllers based on an agreed-upon protocol. The controllers send their readings to the Data Node to coalesce the readings and standardize them for transfer into a local database and for processing in the Analytics Node.

The Analytics Node takes these inputs from the Data Node and uses this input data to analyze the effect of actuation decisions on controlled environmental conditions, and their effect on crop health. It Node provides its output to the Decision Node, which makes decisions about adjusting the controlled environment, such as modifying a heating element, maneuvering a robotic aperture device, and adjusting sensor configuration. In the following, we discuss different subsystems/components of the proposed architecture in detail.

A. Data Node

In the Data Node, sensor readings sent from their respective controllers are aggregated and cleaned up: nil or null values are excluded, and erroneous readings are removed from the set. The node takes, as input, K sets of $(N \times 1)$ vectors of output readings from the sensor controllers in the system, with K being the number of different sensor controllers. Data is packaged into a nested JavaScript object notation (JSON) format, with readings indexed by the type of sensor (i.e., CAN, I^2C, UART). Readings are aggregated by either including each sensor value or by averaging values of the same type (e.g., temperature, humidity) before validation. After the multiple readings from all controllers are validated and combined into a single JSON document, then they are sent over the LAN to the Analytics and Database nodes in transmission control protocol (TCP) packets. The packets consist of a vector of size $(K \times N) \times 1$ containing the coalesced readings. The Analytics Node then acknowledges receipt of the data packet before the Data Node can aggregate more readings. Any subsequent readings received by the Data Node are kept in a buffer pending processing. Once the buffer has reached maximum capacity, the sensor controllers are signaled to halt sending their packets until a certain amount of time has passed. The backoff time increases with subsequent expirations of time counters following a binary exponential backoff (BEB) model from network engineering. In BEB model, once a failure (in this case, halt signal) is detected, the amount of waiting time for the next attempt is doubled. The proposed model translates the Data Node's halt signal to the controllers as the failure [5]. Thus, consecutive halt signals induce twice the delay of their predecessors to relieve backlog in the node. The controllers stop reporting readings until the delay time is completed, after which they resume sending until they receive another halt signal.

B. Sensor Controllers

One sensor controller exists for each different protocol that the sensors use. Each sensing instrument is directly managed by its respective Sensor Controller, which makes configuration changes and polls sensors for their data readings. New sensors are registered with the system before being able to communicate. Sensors are polled, according to their available sampling rates and the experimental requirements, for their raw readings by the controller. Power consumption is

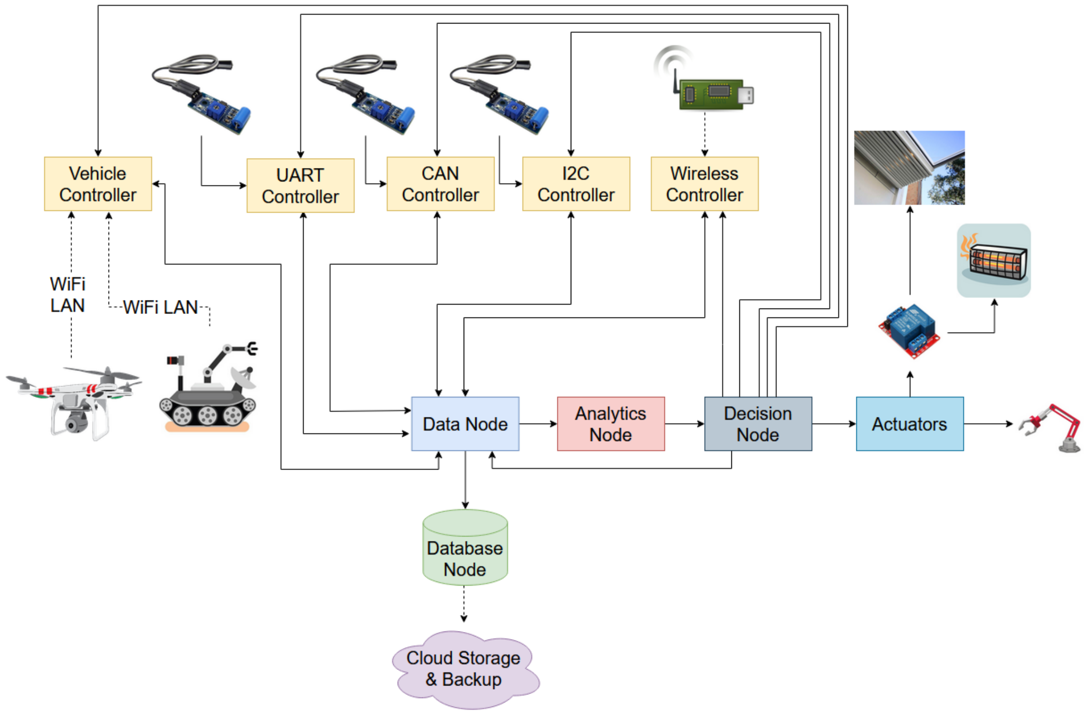

Fig. 1. Proposed architecture for controlled-environment agriculture.

minimized by using the appropriate sleep policy in each sensor controller. Up to N sensor readings from devices connected to the controller are taken as input and reduced to a $(N \times 1)$ vector output. Reduction is typically an average computation on all readings from the managed sensors but the exact function is specified by the experimental requirements and can be `average`, `max`, `min`, or `sum`. Each Sensor Controller parses configuration instructions from the Decision Node and ensures they are valid commands before transmitting them to the specified sensors. Any configuration changes requested by the Decision Node are carried out by the individual controllers directly without requiring the other controllers to receive any extraneous messaging.

C. Wireless Controller

The Wireless Controller acts as a specialized Sensor Controller and is responsible for configuring and communicating with connected wireless sensors. Each sensor acts as a transmitter, where it reads data and transmits it back to the controller for reading. The controller itself is a transceiver that reads the data as well as sends configuration changes requested by the Decision Node to the affected sensor units. The wireless controller is also responsible for sending the data to the Data Node for coalescing and further transmission. Communication is done on the standard 2.4 GHz wireless frequency band; nevertheless, sensors could be programmed to operate at a lower frequency for power savings. Sensors follow a previously proposed adaptive sleep methodology based on network conditions to reduce as much power as they can [6].

D. Vehicle Controller

The Vehicle Controller communicates with and manages the autonomous vehicles in the system. Detected vehicles within the range of WiFi or Bluetooth signals are managed by creating a subnet-like architecture. Robots connected to the infrastructure can automate crucial tasks within the farm and automatically report back via the Vehicle Controller. The controller itself acts as the router for the subnet. To prevent communication issues, the controller follows the BEB algorithm as described in Section III-A. Failures to communicate with a vehicle are recorded, and the controller attempts to periodically reconnect the vehicle. More complex data types, such as image or multimedia data, are processed at the node with functions like object detection, annotation and labeling.

E. Database Node

The Database Node receives the aggregated data in a nested JSON format from the Data Node. It takes the $(K \times N) \times 1$ vector of readings as input from the Data Node, and outputs a single boolean denoting success or failure from interacting with the database. Information in the database is directly controlled by the Data Node, that is, data is cleaned up, formatted, and coalesced before being transmitted for insertion. Any sanitization occurs in the Database Node to deal with unusual nonprinting characters, any irregularities not caught and handled by the Data Node, or problems that would inhibit insertion into the database. A NoSQL schema is used for implementation due to being plug-and-play with different numbers and types of sensing components. Periodic backups would be uploaded to a cloud-based database once per day at a minimum. Data is archived for recordkeeping and for the potential use in future machine learning model training and tuning.

F. Analytics Node

The Analytics Node receives the data readings from the Data Node, in JSON format. Users instruct this node of the functions to perform while the robots and other automaton execute the tasks without requiring human interaction. Before processing can occur, the JSON data is extracted and placed into a vector format. An ANN is located at this node that takes the vector data input and performs the analysis to provide insights into the relationships between different controlled-environment and actuators' settings. Mathematically, the output of a single neuron of an ANN with n neurons in the previous layer can be expressed by:

$$y = f\left\{\left(\sum_{i=0}^{n} w_i * x_i\right) + b\right\}, \qquad (1)$$

where each input has a weight factor w_i that represents its perceived influence on the output, and a bias b is added to the outcome to shift the result towards a more accurate and flexible model. The overall function on the input, f, is the activation function that limits the output of a neuron to specific ranges; typically, the `sigmoid`, `hyperbolic tangent`, `rectified linear unit` (RELU), and `softmax` functions are used. The model is trained and tuned based upon previous readings recorded by the Data Node. Most of the training data comes from archived data, while tuning is done periodically with live data for improvements. Data from previous harvest years can be used to train the neural network to both predict maximum yield and maintain the conditions to achieve the best results for that field in real-time.

Based on the input data, the Analytics Node conducts analysis of various actuator settings and their effect on controlled-environment variables (e.g., temperature, humidity). The Analytics Node also classifies the controlled-environment conditions in different levels (e.g., severity of frost, temperature stress, humidity, etc.) based on the observed sensor data and actuator settings. The Analytics Node also provides predictions on the effects of different actuator settings, which is provided as input to the Decision Node to assist in decision-making. Results from the neural network can help identify important factors affecting crop health and quality, and help improve them through constant adjustments that are made by Decision Node based on the analysis from the Analytics Nodes. The system periodically performs this cycle of sense-analyze-direct at a defined time interval to let the environment properly respond to the prior changes (e.g., actuator settings) and help prevent noisy and erroneous readings from affecting the system.

G. Decision Node

The Decision Node receives the analytics from the Analytics Node, and take decisions, such as sensor and actuator configurations. The decision node produces generic commands, such as "increase temperature". These commands are analyzed to extract the meaningful configuration changes that need to be made and convert them to the appropriate syntax for each respective controller and actuator. Bounds checks and other data validity tests can be performed in this subsystem. The Decision Node sends each specific configuration change to the appropriate sensor controller and actuator as a packet which is then consumed.

H. Actuators

Actuators can be any device ranging from a robotic aperture used to adjust shutters for sunlight level modulation to a propane heater for controlling temperature. Actuators receive the commands from the Decision Node and alter the relays connecting electrical components or adjusts the apertures (controls) within the system to govern the environment. For example, propane heaters, motorized blinds, and LED lighting systems can be controlled to affect the environment in different ways. Additionally, temperature systems, sunlight controls, and some irrigation systems can be controlled to further automate and regulate the system.

IV. Applications

The proposed architecture for automated field-based tents for CEA allows for a variety of experimental uses. A few of the applications of the proposed architecture are discussed below.

Controlled-Environment Agriculture: Controlled environments where a single environmental condition is analyzed for its effect on crops has been accomplished on wheat and rice in prior works [7]. The proposed architecture enables control of multiple environmental conditions in tent-based field-like settings. Simulations of different weather conditions can be performed with the appropriate sensing and actuating equipment. The decision center of the architecture can be programmed with the ideal environment settings for a given experiment. The system is able to automatically regulate a large, controlled environment that is configurable in software to experimental requirements. Expansion is supported with each sensing controller accommodating several sensors attached and the ability to add any new sensor protocol controllers to the system. The tent system can be used for any crop or plot system as it only depends on sensor readings and agronomic knowledge of the experimental conditions.

Crop Phenotyping: Crop phenotyping applications can benefit from the proposed system. Different sensing components and subsystems can be integrated easily into the architecture by the design of the sensor controllers, which would allow multiple phenomenon to be sensed. The architecture provides custom data logging capabilities tailored to the designer's or user's needs, and allows interchangeability of components to best suit the experiment. Paired with the Database node, collecting and analyzing physiological phenomenon are streamlined, autonomous, and are non-invasive to the crop itself if appropriate sensors and actuators are chosen and installed within the operating area. Image acquisition can be done by a UAV and processed by the Vehicle Controller to capture the interaction between crop and environment [8]. Crop growth stages can be identified via images and paired with the environmental readings for further insights into physiological effects.

Precision Agriculture: Precision agriculture can benefit from an automated, self-learning system in tasks ranging from variable-rate nitrogen/nutrient application (VRNA) to yield prediction improvement. Asebedo and Mengel confirmed in their studies on algorithms for optimal Nitrogen application amounts that more data was required for sufficient model accuracy [9]. Nitrogen application is a crucial step in the farming process for optimal crop growth, yield, and quality. Previous agronomic research has overwhelmingly shown that VRNA is the optimal method for crop fertilization as compared to blanket nitrogen application within a plot. Fields are thus sub-divided, or real-time algorithms are developed, to apply differing levels of nitrogen in different sub-fields. The primary statistic used to determine the optimal nitrogen amount to apply to each section, Red normalized difference vegetation index (NDVI), is a singular data point. While being one-dimensional, tremendous success has resulted from analyzing and categorizing plots based on Red NDVI thresholds. However, other data values, such as weather conditions, soil information, infrared and thermal imaging may not be considered in many real-time calculations and models, which the proposed architecture provides integration possibilities for. For example, in winter wheat, the tillers, head size, and grain filling are used to determine the yield and occur at different points in the Feekes scale [10]. The Feekes stages 1 to 5 occur when the plant is below the soil level, which mitigates pest and other above-ground environmental issues. At Feekes 7, Nitrogen mineralization is easier to assess and evaluate, but has the potential for tiller abortion due to Nitrogen stress [9]. Each stage can be categorized and evaluated on a visual basis. An immediately straightforward solution towards reducing Nitrogen stress is to examine and identify the plant growth stages using the imagery available and incorporate these into the real-time model to improve Nitrogen decisions. Temperature and day length are determining factors for how quickly the crop moves through the stages. Sensing instruments for these values can be easily integrated into the system, and relationships between them can be determined by the neural network within the Analysis Node.

V. Case Study and Experiments

We have developed and implemented a simplified version of the proposed architecture to study the effects of high nighttime temperatures (HNT) on winter wheat in Kansas [11]. The experimental thermostat controller system is installed in custom designed tent structures that are 9.1 meters wide, 14.6 meters long, and 4.4 meters tall. Each structure is placed over eight blocks of 40 rows of different winter wheat varieties. The architecture is simplified and split into three pairs of tents, with one control and one heat tent per pairing as depicted in Figure 2. Heat tents consist of a propane heater connected to the controller, a Raspberry Pi model 3B, via relay, which converts the 5V input signal from the controller to the 250V AC required to engage the heater, along with six

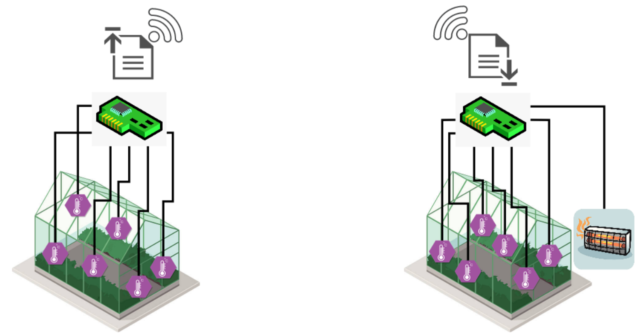

Fig. 2. Pairwise tent system. The tent without a heater transmits its average indoor temperature reading to the heated tent for decision-making.

MCP9808 temperature sensors [12] and one MHZ-19 carbon dioxide sensor [13]. Control tents contain a Raspberry Pi model 3B and the same sensing equipment. Their purpose is to monitor and record the ambient temperature for later use in the heat tent decision-making.

Each pair is connected over a wireless LAN and managed by the control tent, which acts as the router. Each tent records the averaged indoor temperature and the carbon dioxide reading. The control tents transmit their readings to the heater tent for analysis, which are treated as the outdoor temperature. The heat tent, upon receipt of the readings, averages its recorded temperature and compares it to the received readings: above a 4°C threshold causes the controller to disengage the relay, and below 4°C engages the relay. The goal is to maintain a 4°C warmer temperature in each heat tent than the control tent temperature for the duration of the experiment. The system is programmed to read temperature data, decide on the relay state, log sensor and actuator data to file, and sleep. This cycle repeats once per minute to allow the change in relay state to propagate through the environment.

Upon startup, the system detects the type of sensors connected and initializes the main controller for interfacing with the thermostat system. Two options for the main controller exist for the system: one for the heat tents, and one for the control tents; the system, at boot, selects the correct one. Each controller initializes the sensors, reboot error counter, and initiates system logging for diagnostic purposes. Once initialization is completed, the main logic loop begins with calibrating the CO_2 sensor to the current environment. Next, the MCP9808 sensors that were detected previously are polled on the I^2C bus by address with the exclusion of user-defined reserved addresses for the clock module and erroneous addresses. Any errors that occurred are logged to file and the error counter is incremented. Once a programmed level is reached with the error counter, the system automatically restarts itself to resolve the hardware issue. A programmed number of maximum restarts prevents the system from continuously restarting in order to keep the controllers online for sending data and to maintain the HNT environment on the crops even if the temperature threshold is exceeded. The temperature of the heat tents is compared to the control tents and used to determine the relay status, which is engaged by the

Raspberry Pi's GPIO pins. Each step in the process is logged for troubleshooting and archival purposes.

Both the heat and control tents act as a fusion of several of the components within the proposed architecture. They contain the following components, which are present indirectly in software as code chunks or Python classes: Data Node, Analysis Node, Decision Node, and Sensor Controller. Each tent reads the data from their sensors. In our experiments, the Sensor Controllers do not send configuration commands to the sensors, instead the default configuration and error resolution mechanisms are employed to a satisfactory level. Control tents send their data to their heat tent partners for analysis, similarly to the Analytics Node. There is no neural network present, and the decision was a choice based on a linear model, carried out by the controller itself. Data is logged to an Excel spreadsheet file and not imported into a database. The copies of the data are stored locally via replication on several different disk drives. Instead of a dynamic sleep schedule, a static minute of sleep is used in our experimental setup. The selected sleep interval is sufficient to allow the heat change to propagate through the system. The experimental implementation does not have the capabilities for image and video processing. Additional computing hardware is required for these tasks, and a redesign of the pairwise architecture is recommended.

In our experiments, the system was able to achieve a controlled environment of, on average, 3.773 °C warmer than the control areas for the duration of the heat stress as depicted in Figure 3. The three respective heat tents (H1, H2, H3) were analyzed independently. Heat loss was primarily due to gaps in the plastic walls used to seal the tents. The inaccuracies in sensing implements also occurred but were within the manufactured tolerance of 0.25 °C. Heat tents 1 and 3 achieved greater than +3.8 °C temperature while heat tent 2 exhibited +3.6273 °C temperature primarily due to early relay activation issues that were resolved with replacement hardware. Heat tent 2 would engage the relay, which in turn was supposed to turn the heater on. However, the relay was faulty and would engage only a fraction of the time. This caused the heater to remain offline for extended periods and not reach the 4 °C threshold required to maintain the heat stress. After identifying and diagnosing the fault to the relay, replacement relay sets were installed, and the issue was resolved. Additional results and data are available online regarding the system's performance measured by the crop response [11].

VI. Conclusions

While sensor systems design in agriculture are sufficient for individual, specialized experiments, there is a growing demand for an overarching architecture that is versatile, and can observe, analyze, and control multiple environmental conditions. Agronomic experiments require, and rely on, a high quantity of data that current experimental architectures are not equipped to collect, manage, and analyze in real-time. Agricultural models can be improved with additional data and sensing implements and require a solidified architecture and framework to succeed. The proposed design packages components based upon role and can accommodate a variety of sensing instruments, actuators, and autonomous robots with little effort. The proposed architecture can be utilized for a variety of applications, such as controlled-environment agriculture in field-like settings, crop phenotyping, precision agriculture, and experimentation for agricultural and agronomical research. We have developed and deployed a prototype of the proposed architecture, which demonstrates that the proposed architecture can monitor and control the nighttime temperature stress as well as other environmental conditions for winter wheat.

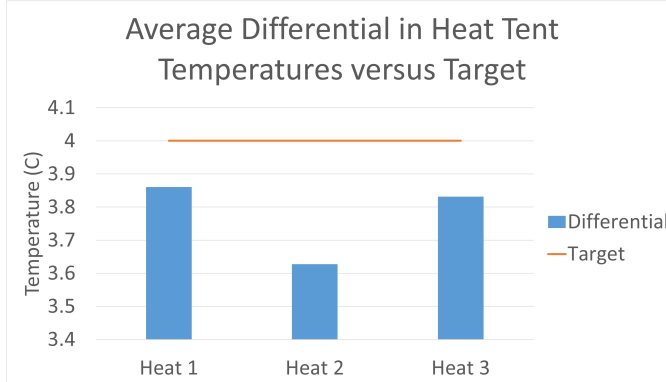

Fig. 3. Average recorded temperature differential in heat tent combinations.

References

[1] Cornell University, "About CEA," cea.cals.cornell.edu. [Online]. Available: https://cea.cals.cornell.edu/about-cea/. Accessed: Mar. 20, 2021.

[2] GLASE, "Lighting, Sensing and Controls" glase.org. [Online]. Available: https://glase.org/benefits/lighting-sensing-and-controls/. Accessed: Mar. 21, 2021.

[3] A. Castañeda-Miranda and V. Castaño-Meneses, "Internet of things for smart farming and frost intelligent control in greenhouses," Computers and Electronics in Agriculture, vol. 176, September 2020.

[4] M. Farooq, S. Riaz, A. Abid, K. Abid, and M.Naeem, "A survey on the role of IoT in agriculture for the implementation of smart farming," IEEE Access, vol. 7. pp. 156237-156271, Oct 2019.

[5] Arvindpdmn Sangeetha-Prabhu, "Binary exponential backoff," devopedia.org. [Online]. Available: https://devopedia.org/binary-exponential-backoff. Accessed: Feb. 15, 2021.

[6] G. Anastasi, M. Conti, and M. Francesco, "Extending the lifetime of wireless sensor networks through adaptive sleep," IEEE Transactions on Industrial Informatics, vol. 5, no. 3, pp. 351–365, Aug 2009.

[7] A. Chaturvedi et al., "Elevated CO2 and heat stress interactions affect grain yield, quality and mineral nutrient composition in rice under field conditions," Field Crops Research, vol. 206, May 2017.

[8] M. Tariq, M. Ahmed, P. Iqbal, Z. Fatima, and S. Ahmad, "Crop phenotyping," Systems Modeling, pp. 45–60, 2020.

[9] A. Asebedo, "Development of sensor-based Nitrogen recommendation algorithms for cereal crops," Ph.D dissertation, Kansas State University, Manhattan, KS, 2015.

[10] E. Lentz, P. Paul, and L. Lindsey, "What is the meaning of Feekes growth stages in wheat," acrops.osu.edu. [Online]. Available: https://agcrops.osu.edu/newsletter/corn-newsletter/what-meaning-feekes-growth-stages-wheat. Accessed: Apr. 4, 2021.

[11] N. Hein et al., "Improved cyber-physical system captured post-flowering high night temperature impact on yield and quality of field grown wheat," Scientific Reports, vol. 10, no. 1, Dec 2020.

[12] Adafruit, "MCP9808 high accuracy I2C temperature sensor," adafruit.com. [Online]. Available: https://www.adafruit.com/product/1782. Accessed: Apr. 3, 2021.

[13] N. Vasilchuk, "MH-Z19 CO_2 and temperature sensor," esphome.io. [Online]. Available: https://esphome.io/components/sensor/mhz19.html. Accessed: Apr. 3, 2021.