Agricultural Automation TestBed

Virginia Tech ME 5735

Christopher Jelesnianski
Electrical & Computer Engineering
Virginia Tech
Blacksburg, USA
Bielsk1@vt.edu

Jackson Klein
Mechanical Engineering
Virginia Tech
Blacksburg, USA
jklein3@vt.edu

Christopher Kappes
Mechanical Engineering
Virginia Tech
Blacksburg, USA
ckappes@vt.edu

Peter Racioppo
Mechanical Engineering
Virginia Tech
Blacksburg, USA
rpeter8@vt.edu

Abstract *— This paper presents progress on an Agricultural Automation Testbed, for the purpose of continuously monitoring plant movement. Previous work had produced a rudimentary testbed capable of motion in 3 dimensions, but which needed continuous supervision. This work takes this testbed as its basis and modernizes it, revamping its controller, interfacing, and data-collection mechanisms such that the system is capable of remote and autonomous control. The completed system is ready for use and can easily accommodate additional features.*

Keywords — agriculture; automation; microcontroller interfacing; Internet of Things; remote control; GUI

I. Introduction

With the miniaturization of computing technology over the last decades, the use of embedded systems which automate mundane or difficult tasks has become increasingly commonplace. Danville, Virginia, previously an agricultural hub, has faced widespread unemployment and impoverishment due to a loss of agricultural jobs, largely as a result of these trends. The Institute for Advanced Learning and Research (IALR), created to perform agricultural and educational research, to address these problems, is investigating whether the small but discernible movements made by plants can be used to actively monitor their health. Previous studies of plant movement and of plant phenotyping have shown that a drop in activity may indicate that a plant is reallocating resources to fight infection or agricultural pests. [1, 6] Employing computer vision techniques may allow for a quantitative assessment of plant health, which could prove useful in large-scale agricultural enterprises. [2, 4] In fact, automated test beds have begun to be used to aid in the monitoring of plants, including one in which plants are rotated underneath a stationary camera and sensor system. [3]

To further investigate the hypothesis of a link between plant health and movement, and to improve on plant monitoring systems, the institute created the Agricultural Automation Testbed (AgBed), a facility for growing a number of small plants in a bed below a track equipped with a moveable camera. This system employs motors in a CNC type arrangement to move a head equipped with a camera, and eventually other sensors, on a gantry over the top of stationary plants. The testbed allows the user to take multiple images of the enclosed plants over an extended growing time period. This system was, until now, run manually, hampering the acquisition of a large amount of data due to the inaccuracy of images. To acquire data, a CNC automated program was used to move the head while a separate computer with nothing more than a timer was used to take photos of the plants, using a webcam mounted on the head. A significant issue was present, in which the camera's photographic timing and the CNC machine timing became mismatched, causing pictures to become misaligned from their intended targets. It was thus necessary to sort out which pictures mapped to certain plants on the bed, and lead to tests being redone due to difficulties matching plants to their photos. On top of this issue, due to the sensitive nature of photographing small movements in plants, reproducibility in camera positioning is vital in order to reduce artificial movement noise induced from relative camera movement between successive photographs. To address these problems, this paper describes updates made to the testbed to allow automated operation, including the development of a microcontroller based control system and a graphical user interface. The aim of this project was to allow the camera and CNC movement to be controlled through one main hub, ensuring images and movements of the testßbed line up accurately and consistently for easy data analysis.

II. Project Overview

Improvement of the AgBed facility required the development of a robotic gantry system capable of motion in three dimensions, appropriate electrical systems for driving

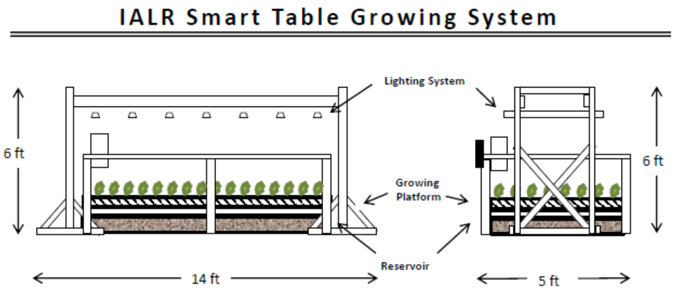
This project was sponsored by the *Institute for Advanced Learning and Research (IALR)*

each of the three motors, an update to the camera and associated camera controls, and an easy-to-use computer interface to direct the position and operation of the camera. In addition, the system needed to be capable of autonomous operation over a long period of time and recalibration in the event of errors in camera position. The development of each of these sub-systems is described below.

III. System Design

A. Mechanical Design

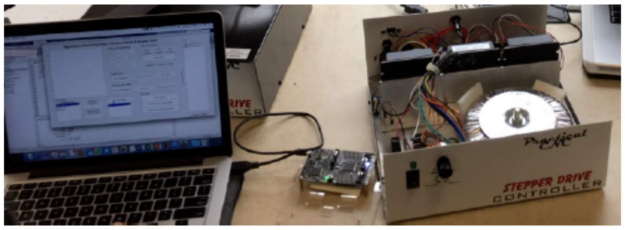
The AgBed utilizes a smart table growing system, equipped with lighting and a water reservoir, which is enclosed by a Computer Numerical Control (CNC) router track. Three stepper motors mounted on the tracks drive a camera equipped gantry in the x and y directions via pulleys, and in the z direction via a ball screw, allowing precise positioning over plants. Figure 1 shows a conceptual layout of the AgBed while Figure 2 depicts the actual AgBed, on location in Danville, VA.



1. Conceptual diagram of AgBed setup

2. Photo of AgBed setup

Before the redesign presented in this paper, the stepper motors were provided power and pulse width modulation (PWM) signals by a stepper motor controller, which draws AC power from the wall and converts it to DC via a large transformer. The controller used outdated peripheral interfacing (RS-232 and DE-9 Serial lines) and the control software provided was intended to be used for a regular CNC router, with support being no longer available. In addition, the original AgBed required three computers: one to interface with the CNC controller, another to interface with a low-resolution webcam, and the third to record data. All experiments were performed manually, which made experimentation a tedious and time consuming process.

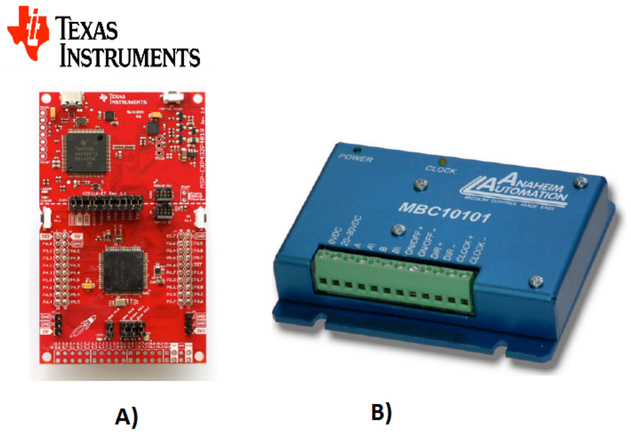


3. Photo of electronics setup

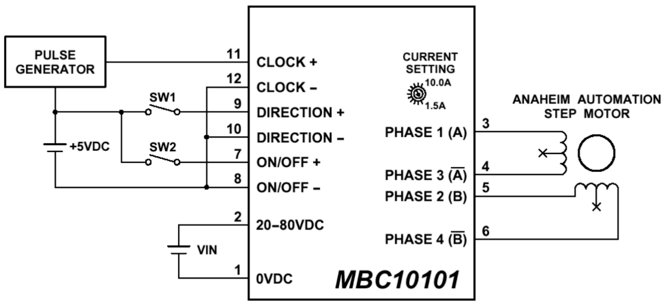
The system implemented in this work has been consolidated such that only a single PC and microcontroller are needed per AgBed, as pictured in Figure 3. The following sections describe the electrical and computer interface design of the new system.

B. Electrical Design

A Texas Instruments MSP 432 microcontroller (Figure 4A) was chosen to send movement commands to the Anaheim Automation motor drivers (Figure 4B), already within the stepper motor controller. Figure 5 displays the needed circuit diagram to properly supply power, PWM, and on/off and direction information to each of the three motor drivers.



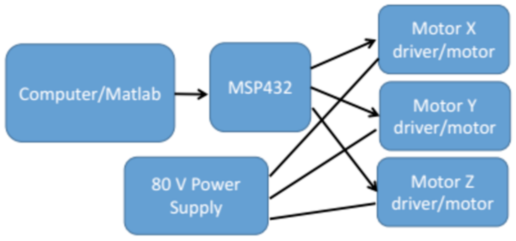
4. A.) MSP 432 Microcontroller [6] B.) MBC10101 Motor Driver [7]



5. Motor driver circuit diagram [7]

As in the original design, the motor drivers draw power (up to 80 VDC) through lines 1 and 2. Lines 7-12 are replaced by outputs from the MSP 432, including the PWM on line 11 and the on/off and direction data on lines 7 and 9, respectively. The

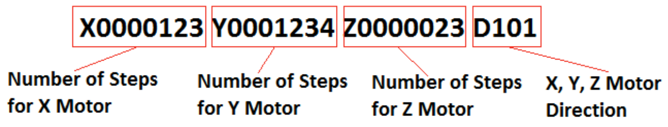
5 VDC source is replaced by the 3.3 VDC rail on the MSP 432. The low quality web cam previously employed was replaced with a 1.4 MP GigE camera, connected via Ethernet to the computer interface. A high-level scheme of the full electrical setup is shown in Figure 6.



6. Electrical setup flow chart

A. Controls

The MSP 432 communicates via UART with a graphic user interface (GUI) run in MATLAB. A formatted data packet specifies the direction and duration of motion for each motor, as shown in the example below:


X0000123
Y0001234
Z0000023
D101
Number of Steps for X Motor
Number of Steps for Y Motor
Number of Steps for Z Motor
X, Y, Z Motor Direction

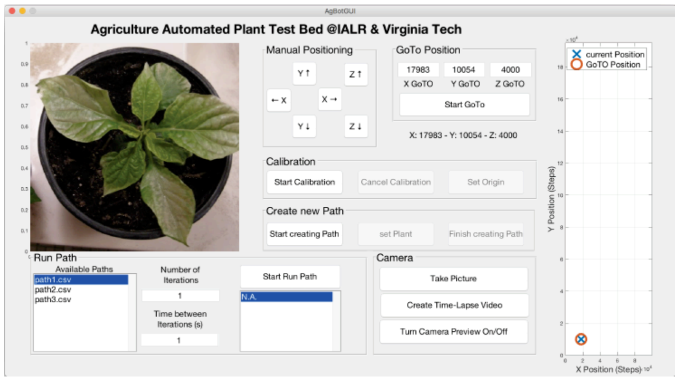
The number of steps to take for each motor is given by three strings of seven digits, one digit for each decimal place. Direction is specified by the last three values, where "1" is used to order clockwise (positive) movement and "0" orders counterclockwise (negative) movement. Seven digits is enough to command on the order of 10 meters of movement of the stepper motors, more than sufficient to cover the extent of the AgBed.

B. Software Design & Interfacing

In the design phase, several requirements were given for the Graphical User Interface (GUI) the scientists would be working with daily. It was indicated it would be useful for the GUI to include the following features:

- A preview function to see how things look at the current location, for ease of alignment and fine tuning of waypoints.
- The capability to initiate an automated run of the AgBed such that no human intervention is needed.
- Manual control (movement and sensor control)
- Calibration to account for motor slippage and other unexpected effects.
- The capability of being updated such that adding or interchanging different sensors is possible in future iterations.

Figure 7 shows the current graphical user interface.



7. AgBed GUI

The following sections discuss each of the panels in the GUI in greater detail.

1. Manual Positioning

This panel enables manual actuation of the gantry as well as camera control. The "Manual Positioning" panel allows one to move the AgBed gantry system in the *x*, *y*, and *z* planes. The head can be moved in one direction at a time for as long as desired by the user by clicking once on the X, Y, or Z buttons to initiate movement. Clicking a second time stops the movement.

2. GoTo Position

This panel provides an alternate means of positioning the camera. The user can input the coordinates of a specific plant in terms of motor steps in the *x*, *y*, and *z* directions and click the "Start GoTo" button. Behind the scenes, the controller figures out the needed number of steps to resolve an input coordinate relative to its current position. Before initiating movement of the gantry, a quick check is performed to ensure the coordinates are within bounds of the AgBed. If an invalid coordinate is input, a warning will pop-up and ask the user to revise their coordinates to ensure a movement is not initiated which will send the head past its hard stop positions.

Additionally, a graphical representation of the AgBed navigable plane is laid out on the right-hand side of the GUI. This graph shows the current gantry position as well as the new projected position. This is an aid to help with fast navigating of the AgBed.

3. Calibration

The calibration panel allows the user to specify the origin of the coordinate frame used in the GUI. Over a long period of operation, the position of the stepper motor will drift away from the desired position. The GUI allows the user to set and reset their origin as needed. The panel works as follows:

- Clicking "Start Calibration" tells the system that calibration is taking place.
- Clicking "Set Origin," the system denotes the current *x*, *y*, and *z* coordinates as the new origin of the system. The old origin is deleted.
- Clicking "Cancel Calibration" cancels this procedure and the old origin remains unchanged.

4. Create New Path

The AgBed will be used to investigate plants of varying age, size, foliage, etc. To cope with the different space requirements and thus placement of the plants, users of the AgBed will need different automation paths for the various experiments they are doing. Therefore, it is essential for users to be able to set a desired automation path. This panel works in the following manner:

- Clicking “Start Creating Path” tells the controller to create a new path matrix which temporarily stores plant waypoints or points of interest that should be processed.
- The user now can use the Manual Positioning or GoTo Position panels to traverse the AgBed and move to points of interest.
- When a point of interest is reached, the user clicks “set Plant”. When this occurs, the controller registers the gantry’s current *x*, *y*, and *z* coordinates and stores them in the temporary matrix. The user then may move to the next desired waypoint and repeat this process.
- After all points of interest have been recorded by the user, the user should click “Finish Creating Path.” When this button is clicked, all points of interest from the temporary matrix are formatted, exported, and saved to an Excel sheet so that the path is accessible for future use in the automated portion of the AgBed.

5. Run Path

The “Run Path” panel allows for users to automatically run given paths automatically, in a “set it and forget it” manner. Once the inputs are selected and the process is initiated, no human intervention is needed until its completion. As an iteration is run, all images captured are listed below the Start button. To begin an automated run, the user can click the “Start Run Path” button. However, the user must set the following parameters beforehand:

- Path to be executed (can be selected from a list)
- Number of Iterations (# of times the given path will be repeated)
- Delay between successive images of the same plant (i.e. the timer starts to count once the first image is taken)

6. Camera

While the camera feature is automated within the “Run Path” panel, it may sometimes be desirable to perform experimental test runs with only the camera. For this purpose, a utility panel was created to perform various camera related task. For now, the implemented features are as follows:

- Capture images manually via “Take Picture”
- Stitch multiple photos together to create a time lapse video via “Create Time-Lapse Video”
- Toggle the Camera Preview On/Off via “Turn Camera Preview On/Off”. The image preview pane is a live feed of what the camera currently sees and is located on the left side of the GUI.

IV. System Evaluation

The authors successfully installed the new system at the two AgBeds in Danville. Full functionality was demonstrated, including manual and automatic movement and identification of incorrect coordinates. Additional testing could include more precise measurements of speed of operation and the amount of incremental error entering the system over time.

While making the system, multiple skills required learning for implementation. Since the authors had not had experience with stepper motors and associated motor drivers, much work was put in to figure out how the system was run. Once the motors and drivers were characterized, the MSP432 was implemented to run the system. This involved learning how to transmit packets of data to and from the MSP432 using UART and a Serial connection, and the optimal way to control stepper motors using a clock signal. Finally, the authors researched how to make a GUI structure in Matlab for communication with the MSP432 and AgBot table, and camera.

V. Conclusion and Future Work

In conclusion, this work presents a modernized approach to performing agricultural automation using microcontrollers. The original system has been updated to require only one PC, while recycling as much of the original infrastructure as possible. In addition, proper interfacing has been created for agricultural researchers using AgBeds so that working on these systems is much easier, less tedious, and intuitive. The GUI interface supports both manual and automatic operation as well as useful features such as a time-lapse video creator.

While much progress has been made, additional customizations which may be implemented are left to the user’s imagination. Both controls and the software design have been implemented with modularity and future updates in mind. The majority of the code written for this project is well documented, so that future changes will be easy to merge with the current version.

It has been observed that some species of plants exhibit purposeful movement in order to achieve an objective, such as reaching sunlight, a water source, or a place to anchor themselves. To better understand plant kinetics, further investigation of this phenomenon using infrared imaging has been suggested. Such experiments would allow researchers to understand plant kinetics by observing the relational byproduct (heat emission) of work performed by plants. To support this investigation, support for an infrared camera would be needed. However, since the current system already supports traditional imaging, adding a different type of camera would be trivial. Additionally, a time-lapse utility already exists to properly document this phenomenon with respect to time.

Similarly, it is obvious that many factors can effect plant growth and development. Factors such as soil moisture, sunlight, temperature, and blights can drastically alter the development of plants. Recent advances have spurred development of sensors with connectivity (Wi-Fi, Bluetooth, etc.), and the so called Internet of Things (IoT). To adequately perform more complex experiments, and thus track more parameters, it will be necessary to add additional modules to the software design. For example, the project could be fitted with a Wi-Fi or Bluetooth receiver to wirelessly transmit data readings at specified time intervals.

An additional goal of the AgBed project is to serve as an educational tool in the surrounding community. This project is intended to help better engage students and expose them to STEM related fields. The AgBed project provides an intuitive and exciting platform to engage students at a variety of educational levels. In the immediate future, a straightforward web portal could be designed to give teachers access to data gathered by scientists at the IALR, for use in the classroom. The project also provides a convenient platform to teach a variety of other subjects, including programming, horticulture, and image processing.

The team would like to thank Dr. Wicks for his guidance and support in this work as well as the personnel at IALR for giving us the opportunity to work on such an impactful project for the Danville community.

VI. CAD Files and Instructions

A. *Files*

https://drive.google.com/open?id=0B9d6qILx-KKtdTZHZzM0T1dlZG8

B. *Instructions for Operation*

1. Initialization

To open the GUI, press the ‘Run’ button (green arrow) in MATLAB (Note: the camera and motor driver box must be connected to the computer).

Three common errors may occur when attempting to open the GUI. If the GUI fails to open, restart MATLAB. If the problem persists, attempt to debug using the instructions below:

1.) The default USB port for the connection to the MSP432 is COM7. If this port fails to close after a connection, MATLAB will output an error about the COMs. If this occurs, enter ‘`fclose(instrfind);`’ into the MATLAB command line to close all ports.

2.) Another less common, but possible COM error is that the port may change from COM7. If this occurs, the new port will need to be identified. In the Start Menu, search for ‘Device Manager.’ Click on ‘Ports (COM & LPT).’ Look for the first port which says UART next to it. If this port is different than COM7, go to the MATLAB GUI code and search for the line: ‘`handles.s = serial('COM7');`’. Replace COM7 with the new port name (COM#).

3.) If MATLAB outputs the error: “*`There are no devices installed for the specified ADAPTORNAME`*,” the camera is not connected. If this occurs, go to the ‘Apps’ tab in MATLAB and open the Image Acquisition Toolbox (top, center). In the ‘Hardware Browser’ window in the top left, right click and then click ‘Refresh Image Acquisition Hardware.’

2. Manual Position

In the ‘Manual Position’ section, the position of the camera may be manually set using the X, Y, and Z buttons. One click turns the motors on, and another turns them off. There are no hard stops on the AgBed table. Be very careful not to run the motors past the limits of the table.

3. GoTo Position

Position can also be set by entering the coordinates into the boxes labeled X, Y, and Z in the ‘GoTo Position’ window. Note that coordinates are given in terms of motor steps and are therefore relatively large numbers (on the order of 10,000). Incorrect entries or entries which will take the system outside the limits of the bed do not run the system and output error messages to the screen. Once a move is finished, the current absolute coordinates of the camera are displayed at the bottom of the ‘GoTo Position’ window. The position of the system is displayed graphically in the figure at the far right of the GUI. The desired position of the system is displayed as a red circle and its current position as a blue X.

4. Calibration

The ‘Calibration’ window is used to zero the system to a desired zero location. Calibration should be performed periodically, as errors in the computed position of the system will add up over time. To calibrate, press ‘Start Calibration,’ manually move the system to the desired origin (the lower left corner of the table is recommended), and press the ‘Set Origin’ button. To cancel calibration during this process, press the ‘Cancel Calibration’ button.

5. Run Path

In the ‘Run Path’ section, the ‘Available Paths’ window displays previously saved paths. Each path consists of the coordinates of each point which the camera will be ordered to move between. Once a path is chosen, the number of times to take the path and the time to wait between runs can be input with the ‘Number of Iterations’ and ‘Time between Paths’ buttons, respectively (default values are 1 and 0). To run through the selected path, press the ‘Start Run Path’ button. The system will autonomously move to each plant location, take a photo at each, and save the photos to file for later use.

6. Create new Path

New paths may be created in the ‘Create new Path’ section. To create a new path, use the ‘Manual Position’ buttons to position the camera over each plant. Adjust the *z*-coordinate until the image is properly focused. Use the live video feed to determine the desired camera position for each plant.

Once the desired coordinates for a plant are reached, press the 'set Plant' button. The coordinates of the plant will be saved file, where they will later be accessed by 'Run Path.' After the coordinates for all plants have been set, press the 'Finish creating Path' button.

7. Camera

Press the 'Turn Camera Preview On/Off' button in the Camera section to begin live video feed. To manually take a image, press the 'Take Picture' button. Once a group of pictures are taken, a time-lapse can be generated using the 'Create Time-Lapse' button.

Warnings

As this is the first prototype of the system, some potential problems should be noted and eventually addressed:

1. While there are some safety stops put into the GUI to not allow input movements past the boundaries of the table, if the origin is set to the wrong position, there is the potential for a move to be commanded which will run past the end of each side. There are simple hard stops on the system, but these do not contain a switch to stop the motor movement, so motors can be burned out, or gear systems stripped. A fix for this could be an implementation of a CNC type hard stop limit switch to turn off all motors when touched.
2. There are no implemented safety stops when using the X, Y, and Z manual buttons. Please use caution when using this functionality. Remember: one click starts movement, and a second click stops movement. Again, limit switches as mentioned above would prevent problems from occurring.
3. As currently implemented, there is no position feedback from the stepper motors themselves. We are simply counting how many steps were commanded to be taken. Thus, due to backlash in the gearing and track, inaccuracies are possible and precision may be gradually lost after many runs of a given path. It is recommended that the origin be reset after each full test (many runs of the same path). A rotary or linear encoder could be implemented on each axis to check movement accuracy. Alternately, a dot could be placed on each plant, and computer vision software could be implemented to adjust the picture placement in each frame of a video made of the images so the plant is in the same position throughout the video.
4. To set the origin easily each time, it is recommended to use a corner (like on a plastic part) which can be placed in the same position each time, and use the preview to place the camera in the same position each time an origin is set.
5. Wires connected to the MSP 432 Launchpad should not be moved from the pin on which they are placed, as these are uniquely identified within the software on the Launchpad itself.
6. The power supply for the motors may be as high as 80 V, and each motor may draw up to 10 Amps when in use. DO NOT TOUCH WIRES WHILE IN OPERATION! A future step should be to clean up the housing for the MSP 432, motor drivers, and power supply.
7. Like mentioned earlier, the system is a prototype. Do not let unadvised people operate the system. Never let the system run unattended. Be aware of the emergency and power stops.

VII. References

[1]. N. Fahlgren, M. Gehan, I. Baxter. *Lights, camera, action: high-throughput plant phenotyping is ready for a close-up.* Current Opinion in Plant Biology, 2015.

[2]. D. Oosterhuis, S. Walker, J. Eastham. *Soybean leaflet movements as an indicator of crop water stress.* Crop Science, 1985.

[3]. R. Furbank, M. Tester. *Phonemics - technologies to relieve the phenotyping bottleneck.* Trends in Plant Science, 2011.

[4]. M. Kacira, P. Ling, T. Short. *Machine vision extracted plant movement for early detection of plant water stress.* Transactions of the ASAE, 2002.

[5]. M. Kacira, P. Ling. *Design and development of an automated and non-contact sensing system for continuous monitoring of plant health and growth.* Transactions of the ASAE, July 2001.

[6]. Texas Instruments, *"MSP432P401R LaunchPadTM Development Kit (MSP‐EXP432P401R)"* Users Guide, March 2015 [Revised July 2016].

[7]. Anaheim Automation, *"MBC 10101 – Microstep Driver"* Data Sheet.

[8]. S. Lowman. *Utilizing beneficial bacterial endophytes to promote switchgrass growth in low-input agricultural production systems.* Doctoral Dissertation, Virginia Tech. 2014.