Development of Open-Source Gantry-Plus Robot Systems for Plant Science research

Adwait A. Kaundanya

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical Engineering

Dong Ha, Chair
Song Li, Co-chair
Haibo Zeng
Kaylee A. South

December 06, 2024
Blacksburg, Virginia

Keywords: Robot Operating System 2, 3D Plant Phenotyping, Multi-Robot System, Moveit2, Motion Planning, Gantry, Manipulator

Development of Open-Source Gantry-Plus Robot Systems for Plant Science research

Adwait A. Kaundanya

(ABSTRACT)

Affordable and readily available automation options for plant research remain scarce, however with the availability of such a system, many research tasks can be streamlined. In this project, we demonstrate a prototype of such an open-source, low-cost, heterogeneous robotic system called Mini T-Rex. We combine two over-the-counter robots and leverage the ROS2 framework to control this heterogeneous system. This system provides a unique advantage of sensor-to-plant method to capture multi-view images at any angle and distance within the workspace. We demonstrate how making a digital twin in ROS2 can help to control a heterogeneous system via abstracted hardware control. We also talk about I2GROW Oasis which is a robotic system consisting of a remotely controlled robot with the ability to capture top-view images. In this thesis we describe the hardware and software design of both these robotic systems. To use this robotic system, the plants can be grown on a growth bed or a hydroponic system below the Mini T-Rex robot, and the camera will approach the plant without any contact with the plants due to the precise control of the robotic manipulator. We used the system to capture several large data sets of 3D phenotypic data for Solanum lycopersicum, Lactuca sativa, and Thlaspi. In conclusion, we have developed a 9-degree of freedom, fully open-source heterogeneous robotic system capable of multi-view, camera-to-plant image capture for plant 3D model reconstruction called Mini T-Rex. We show how to use gantry like robots for phenotyping and create longitudinal datasets by automating these high precision robotic systems.

Development of Open-Source Gantry-Plus Robot Systems for Plant Science research

Adwait A. Kaundanya

(GENERAL AUDIENCE ABSTRACT)

Robotics are being widely used for automating tasks that are monotonous and require high precision. However, developing such application specific robots in itself is a complicated and tedious task. Having different aspects like mechanical design, robot fabrication, software design makes it difficult for any individual or small groups to develop such robots. In order to facilitate plant researchers who may not have any experience in designing robots, we have developed a general robotic system that can be easily assembled and adapted for applications. In this thesis, we discuss how this robotic system can be made using over the counter robots and discuss how the software makes it intelligent enough such that it can navigate the course without any collisions and control the robots as if they are part of one system rather than two different robots that are controlled individually. This enables using the vendor provided software rather than designing the entire robot from scratch. We also show another robot kits, the FarmBot, which can be assembled and adapted to particular use case of monitoring hydroponically growing crops. We demonstrate how this robot can be used as part of complex systems and how it can be automated to collect images to monitor plant growth. We describe in detail of how a user can go from computer aided design(CAD) to hardware control of the robot, and how this system can be used for phenotyping of plants namely early girl tomato, lettuce, and pennycress.

Dedication

Dedicated to my family, friends and teachers.

Acknowledgments

I would like to take a moment to express my gratitude to my advisor Dr.Song Li. His encouragement and trust in me to complete the project have been pivotal in my academic journey. Thank you Dr. Li for your support in my Master's degree.

I would also like to thank Dr. Dong Ha for agreeing to be my committee chair. Thank you to Dr. Haibo Zeng and Dr. Kaylee South for providing me with feedback and being part of my committee.

I want to thank the members of the Li Lab of Applied Machine Learning in Genomics and Phenomics, Sanchari Kundu, Shadab Haque, Chirag Patil, Razan Alojeen, Tran Chau, Manisha Chaudhary, and Krishna Vamsi Dhulipala.

I would also like to thank Peter Forstner for his help in building the I2GROW Oasis and Harish Kannan for his help in SolidWorks.

Lastly I would like to thank my parents for their support in my academic journey and my friends for their support and precious memories.

Contents

List of Figures

List of Tables

List of Abbreviations

CAD	Computer Aided Design
DoF	Degrees of Freedom
LIDAR	Light Detection and Ranging
NFT	Nutrient Film Technique
RGBD	**R**ed, **G**reen, **B**lue, **D**epth
ROS2	Robot Operating System 2
RRT	Rapidly exploring Random Trees
SW2URDF	Tool to convert CAD to URDF files
URDF	Unified Robot Description Format

Chapter 1

Introduction

In recent years, automation and the use of robots to automate processes has been expanding steadily and are evidently successful. One example is the increasing applications of collaborative robotics in areas where the monotonous and repetitive tasks can be covered by the robotic counterparts. Robots can perform complex monotonous tasks with the same quality and can be programmed to run on a fixed schedule with minimal human intervention[31]. Although this trend of automation is evident in most industries, application of robotics in agriculture remains limited in the areas including seeding, harvesting, irrigation. A broader adaptation of robotics and automation has the promise to grow the productivity in more domains in the agricultural industry[3, 13, 35].

In controlled environment agriculture (CEA), the scope of the plant growth can vary from growth chambers in research labs which are usually several square feet in size, to small greenhouses of hundreds of square feet, and to very large greenhouses spanning several acres of land [3, 15, 22]. In these application areas, gantry robots, also commonly known as cartesian robots, have found an ideal fit. These robots can perform autonomous and high precision tasks, at the same time, they are easy to scale to different agriculture application scenarios, and have the flexibility to add application specific attachments. One of the central advantages is that gantry robots are easier to deploy in agriculture settings as compared to unmanned aerial vehicles(UAVs) and ground vehicles(UGVs), which require more sophisticated navigation algorithms. On the other hand, since crops layouts don't move after planting without human intervention, gantry robot is much easier to be programmed to navigate a farm or a greenhouse in a pre-defined layout while performing complex manipulation task without the complications such as stability control for UAVs. Despite the advantages of gantry robots, most are designed and manufactured from scratch, with proprietary software, making it difficult to replicate and scale for other applications. [13, 36]

From the plant research domain, a lot of effort is being put to automate processes that require precision and accuracy. In contrast to industrial applications where high-throughput is highly valued, plant/agricultural research robots have a focus on carrying out precise applications with the ability to be fully dexterous and have the potential to attach different sensors and attachments[30, 34].

In our recent paper [34], we have described the T-REX robot (an acronym for "**T**he **R**obot for Leaf DNA **EX**traction"), which is a gantry like platform with 6 degrees of freedom made with the aim to automate the process of leaf DNA extraction. The overarching goal of T-REX is to perform automatic disease detection using nanopore sequencing (a type of DNA sequencing technology) of the plant microbiome. The T-REX robotic system is capable to grab and cut leaves from a living plant, and to achieve this, the robot first capture multi-view stereo images of a plant to perform 3D reconstruction, and to generate a dense point cloud. This data is used to calculate the leaves that are accessible to a robotic manipulator, and computer vision algorithm was developed to isolate each leaf to obtain the plane of the leaf and estimate its pose. This information is used to guide the manipulator to clamp a microneedle array to the target leaf to extract DNA for the downstream process of using nanopore sequencing to determine the microbiome of the target leaf sample.

This evidently shows how these gantry like robots are quite useful in various scopes in plant research as well as commercial capacity, but a general observation from these is that the system are usually designed from scratch for each application which can pose challenges when it needs to be replicated and slow down the scope of research projects. Even though the software development can be accelerated, the most challenging part is to source hardware materials, fabricate them to the custom dimensions, make frameworks to automate these processes and interfaces for users to use them.

These issues can be mitigated by making an effort towards a general purpose gantry like system capable of handling different kind of end effector attachments along with a robot operating system(ROS2) framework which provides seamless integration and ease in scaling the scope both from the software and hardware point of view. Designing such a system with readily available market solutions can shorten the project timeline and increase the reliability of the system. With this aim in mind, we decide to develop what we call the Mini T-Rex as shown in 1.1. Mini T-Rex is a robot made with a gantry and a manipulator-OpenBuilds 1515 and MyCobot280Pi respectively. The reason behind the name is that we

Figure 1.1: Design of Mini T-Rex in SolidWorks.

want to build a robot that has the capability of all the things T-REX does as shown in [34] but entirely built with over the counter robotic sub-systems and products. Although Mini T-Rex is much smaller in scale when compared to the original T-Rex, it serves as a good proof of concept for such a robotic system completely made with off-the-shelf robots, such that it can be easily adopted by other research groups with limited technical expertise in robotic design.

There are different challenges that need to be addressed before building such a heterogenous system with off-the-shelf robots. One major concern is that, these systems may or may not support ROS2 frameworks depending on the intended application of the standalone system. To build a unified model that represents this hybrid multi-robot system, a software framework is needed for the desired functioning of the robot. Since the robots chosen have controllers of their own, the main challenge is to control these individual homogeneous systems together. This is where developing a ROS2 and MoveIt2 configured package becomes a key factor in achieving this goal.

ROS2 is an excellent framework to build heterogeneous systems because ROS2 was designed to handle such multi-robot systems which may involve different actuating systems and sensors[2, 7]. The communication middle-ware of ROS2 is based on DDS(Data Distribution Service) which is designed for complex distributed services. This enables ROS2 to have nodes running at the target, thus these nodes can be designed for the underlying hardware and still communicate with the whole system in the same language as will be shown in detail in 3.1.2. ROS2 provides real time control and the ability to integrate with real-time operating systems for deterministic control. Using DDS enables the user to choose the style of communication, it can either be publish-subscribe or request-response style of communication which would be highly useful in low power or battery operated embedded systems. With ROS2 the hardware can be completely abstracted from the software simulation and all the communication can happen using the topic where publisher nodes will publish data and the subscriber nodes will use that data to perform the required actions, in our case the MoveIt2 publisher topics also come with timestamps which can be used to synchronize both the systems[33].

In this thesis, we demonstrate how a hybrid system can be controlled using ROS2. One of the robots is natively ROS2 compatible and the other system needs to be configured by making ROS2 configuration packages for it to be ROS2 compatible. In section 3, we will talk in detail about how this was accomplished.

Figure 1.2: FarmBot as used in the I2GROW Oasis project.

We also tested another gantry robot called FarmBot as shown in Figure 1.2 as part of the I2GROW-Oasis system. This system has been developed as part of the I2GROW project. One of the objectives of this project is to develop robotic and sensor systems to determine optimal growth conditions and green house gas reduction for indoor plant production. The I2GROW-Oasis is developed as a phenotyping tool for sensing and measuring growth with multi spectral and 3D sensor for hydroponically grown crops[20].

This system is the first step towards establishing the I2GROW-Oasis testbed, which is a 3-axis linear robotic system with LED grow lights, NFT (nutrient film technique) systems for supply of nutrients and water to the plants, and an RGB camera to take images of the plants at predefined locations at regular set intervals. The entire design of the system and its capabilities is discussed in section 3.2.

1.1 Thesis Outline

This thesis is structured in the following way:

- Chapter 2 discusses the literature review of different aspects of this project.
- Chapter 3 defines the methodology adopted to achieve the aim and the hardware and software components of the Mini T-Rex and I2GROW Oasis.
- Chapter 4 gives the experimental setup and shows the results obtained from them.
- Chapter 5 concludes the thesis by showcasing the proof of concept and the objectives achieved. Outlines potential future improvements.

Chapter 2

Literature Review

In this section, we will review literature on three major topics related to the work in this thesis. First, the motion planning algorithms, particularly for serial robots; second, how plants can be monitored using gantry like robots; and third, techniques for 3D reconstruction, which can be used to measure plant growth over time.

2.1 Motion Planning

Motion planning is the process of determining the joint states that the robot can follow to go from position A to position B. To achieve this, the user has to define the configuration of the starting state and the goal state of the robot. This requires determination of the parameters of the robot which the algorithm can then use to plan the joint states.[17]. Sampling based motion planning is a general choice for robotic applications, this involves sampling random points in the space and then connecting them to form a path without any collisions. This algorithm creates graphs where nodes represent sampled points in the configuration space and edges denote the possible connection between nodes[16]. Sampling based motion has different algorithms that can be used to find the optimal path. One of the widely used algorithms is the RRT(Rapidly exploring Random Trees) and the different adaptations of RRT like the RRT*, Q-RRT, RRT-Connect[12, 33]. In ROS2, the **O**pen **M**otion **P**lanning **L**ibrary(OMPL) planner provides various of these algorithms that can be set by the user for motion planning of the robot.

2.2 Plant monitoring using gantry robots.

Robotic technologies have the potential of enhancing the throughput of plant phenotyping, which is an emerging research field in plant science including applications in measuring crop height [10], determining nutrient content of agriculture products[43], and identifying plant diseases[9, 39, 45]. Various robots of different styles have been developed and tested on different crops which has demonstrated success of robotics in agricultural applications[11, 25]. However, many challenges remains, and has been discussed in detail in [1]. These challenges include adapting to the dynamic environment where the robots operate, scaling robotic system software, adopting a robot which was made with specific design principles for another application, and keeping up with constantly updating open source tools like ROS.

Gantry robots have found application in closed, controlled, confined spaces where the movement requires high precision and have cobotic abilities. These robots have found a niche in plant phenotyping since the ability of the end effector to reach any point in the workspace. In the paper [5], it was demonstrated how an automated pipeline for machine vision was developed and how the gantry robot was manufactured by a vendor which helps the robot to accurately move the sensor around the plant to collect multi-view data. Different attachments can be attached to the gantry system for diverse applications like thermal imaging or hyper-spectral imaging[6]. In another publication, the efficacy of one plant-to-sensor system with a 6DoF(Degrees of Freedom) robotic arm has been demonstrated successfully by showing the 2D and 3D data collected, and how different phenomic traits can be acquired[41].

These data collected by robots can further be used to train computational models for predicting, monitoring, and ready for harvesting stages in plants. With different sensors that can be attached to the robot, the data can be processed to gather meaningful information for agronomic decision making[19]. From these publications, we also noted that slight modifications to robotic systems can add huge value with little addition in software control. For example, the image data collected can be used to train and deploy models for real time monitoring, and these models can be used for real time control by deploying these models to control the light or nutrient input which govern the growth of the plants.

2.3 3D Reconstruction

3D reconstruction is the process of making a 3D model to represent physical properties. There are different techniques that can be used for 3D reconstruction, and each method may require specific sensors that can capture the data required. These factors should be considered when designing the system in order to optimize on sensor selection[23, 44].

Multi-view techniques have been used and proven to be effective in creating accurate digital twin point clouds of phenotypic traits. It involves taking multiple images of an object from different perspectives and then using the method of triangulation to extract depth information.[8]. Multi-view imaging for 3D reconstruction is a promising technique since it does not required calibrated images. There are many computational algorithms that can use multi-view images for 3D reconstruction [24, 42].

There are different types of sensors that can be used to collect the data required for 3D reconstruction. Intel has released multiple sensors that provide the ability to save RGBD data. One of them is the L515[14] which is a LiDAR that also provides the ability to save RGB data. In a technical report paper [21], they found that in controlled areas where the environmental conditions like the ambient light is being controlled, L515 is more accurate and precise, while providing stable and consistent measurements as compared to other sensors. Generally speaking, LiDAR sensors such as L515 can provide precise depth data for 3D reconstruction. Other types of data like RGB and point cloud can also be captured using L515 to use different computational techniques for the reconstruction.[37, 38] The information obtained from LiDAR can also be used with multi view sequence images to further improve the accuracy and reinforce the data collected with other sensors[40].

In addition to the LiDAR data, another method for collecting 3D information involves the usage of stereo cameras. Since the meta data of these cameras is known by design, the 2 images taken have known meta data which can further be used to facilitate 3D reconstructions[18, 34]. For this thesis, we do not focus on making the 3D reconstruction. Our focus is to deliver an automated, open source, over the counter robot to collect the data required for the 3D reconstruction.

Chapter 3

Methodology

3.1 Hardware and Software Setup of Mini T-Rex

3.1.1 Hardware Description

This section will describe the hardware components that are used to build the Mini T-Rex and their specifications. We will also cover the ROS2 configuration and discuss in detail how this multi-robotic system can be simulated and controlled. We will also elaborate on the design decisions of choosing the components and the advantages of using OTC (Over the counter) robots. Figure 3.1 illustrates the structure and components of the Mini T-Rex robot. It is comprised of three major components. The first is the central PC, which controls the entire Mini T-Rex by running the ROS2 nodes that simulate the digital twin of the system and control the hardware. The second component is the OpenBuilds 1515 gantry, originally a CNC machine, repurposed to act as a gantry that holds the third major component, the MyCobot280Pi which is a 6-degree-of-freedom manipulator. Attached to the end effector of this manipulator is the Intel RealSense L515 LiDAR, responsible for capturing RGB and depth images of plants.

The central PC runs the Ubuntu 22.04 distribution and simulates the digital twin and operates MoveIt2-the motion planning framework. MoveIt2 calculates the entire path of the Mini T-Rex, determining the different joint movements required to ensure there are no collisions as

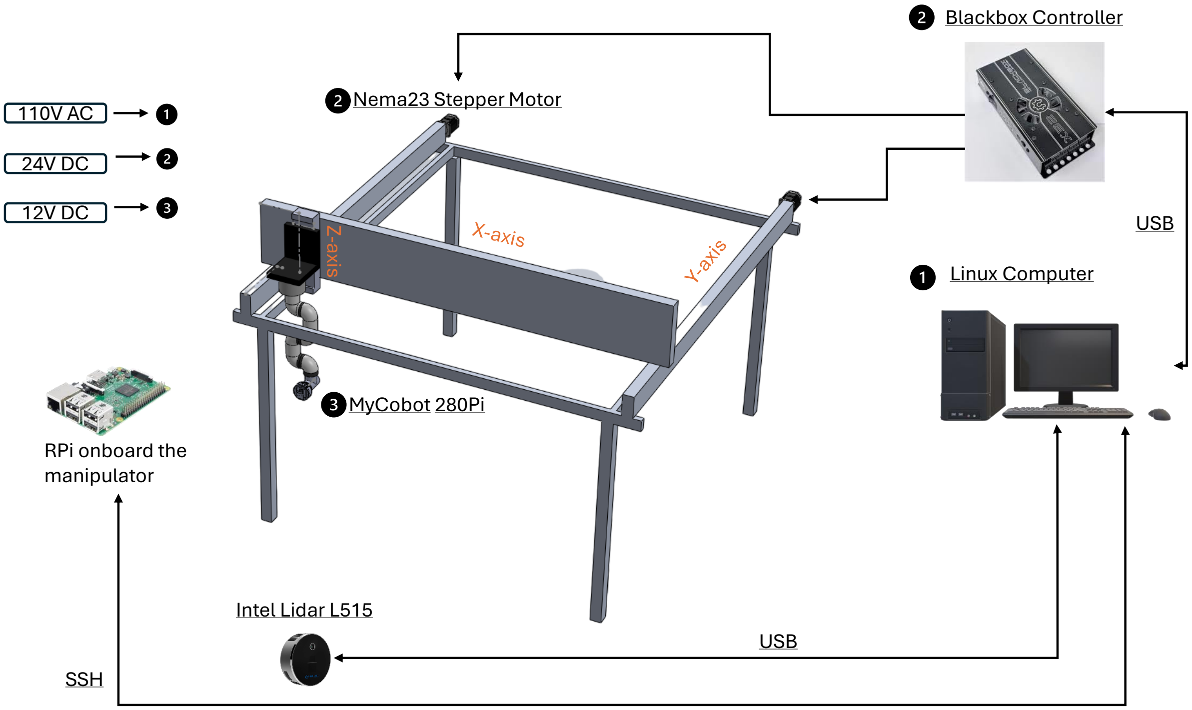


Figure 3.1: System Overview of the Mini T-Rex

Table 3.1: Important individual robot parameters[26, 28].

Robot Parameters	
Robot Parameter	Value
Degrees of Freedom	9
Payload Capacity	250gm
Intel RealSense Weight	∼85gm
Manipulator Weight	850gm
Working Radius of manipulator	280mm
Manipulator Precision	0.5mm
Gantry Footprint	1650mmX1650mm
Gantry Precision	0.05mm∼0.10mm

the end effector reaches the desired position. The central PC is connected to both the OpenBuilds 1515 gantry and the Intel RealSense LiDAR via serial connections. Additionally, the manipulator is equipped with an onboard Raspberry Pi. By establishing an SSH connection to the Raspberry Pi, we can manage it through the ROS2 network as shown in detail in the Figure 3.4 Since all devices are on the same network, we can treat these various components as part of the same ROS2 network, enabling seamless data sharing between them. This is one of the main design changes in ROS2 from ROS where the messaging and servicing layer is the DDS(Data Distribution Service) for low latency, real-time frequent data exchange. The

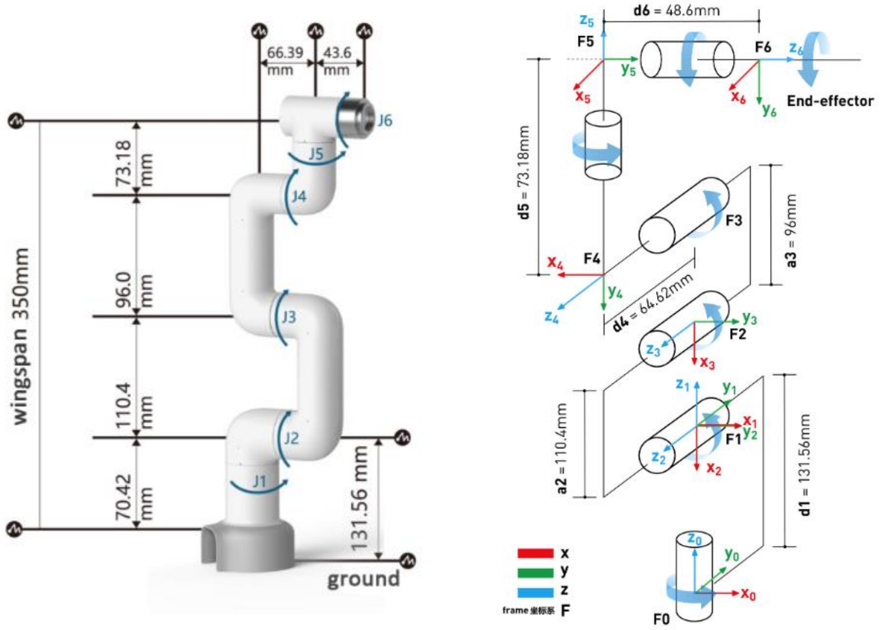


Figure 3.2: Space parameters(Left) and Denavit–Hartenberg(DH) parameters(right) of the MyCobot280Pi.[27]

6-degree-of-freedom manipulator is purchased from Elephant Robotics. Attached to its end effector is the Intel RealSense LiDAR, which is connected to the central PC through a USB serial connection. This LiDAR captures both RGB and depth images simultaneously. It is also capable of continuously collecting depth data in the form of point clouds. To ensure

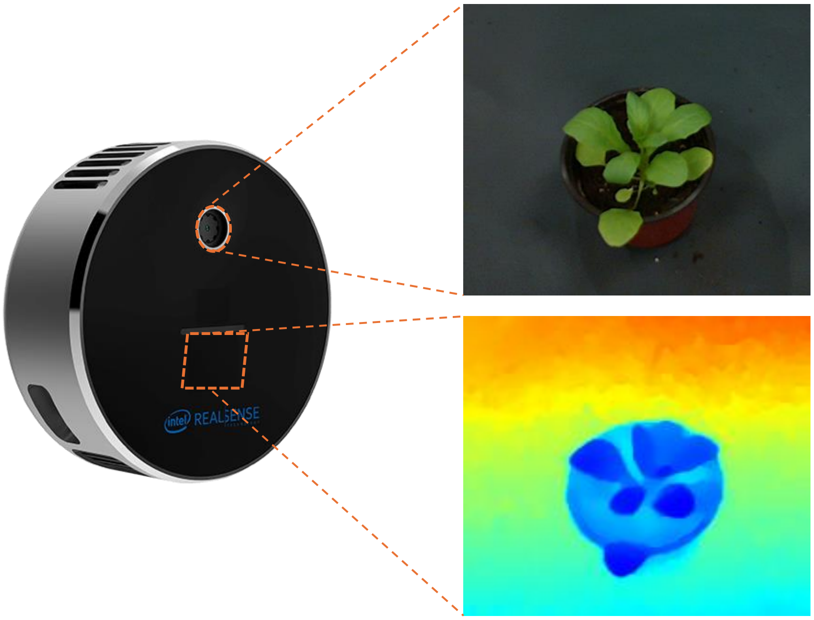


Figure 3.3: Intel RealSense LiDAR L515 used as the sensor.[14]

the proper operation of the Mini T-Rex, it is important to match the dynamic and static parameters of the digital copy of the robotic system. Figure 3.2 shows the static distances between the links and joints, and the figure on the right displays the Denavitz-Hartenberg (DH) parameters we used to model the CAD equivalent of the MyCobot280Pi. This CAD model was then used to write the Unified Robot Description Format (URDF) file of the Mini T-Rex, which takes all the properties required for the calculation of the movement of the joints to reach a desired configuration. Based on these physical properties, the MyCobot280Pi has a working radius of 280 millimeters and a payload capacity of 250 grams, which is ideal for the Intel RealSense LiDAR sensor. All communication with the manipulator is routed through the onboard Raspberry Pi as shown in Figure 3.4. We implemented a subscriber node running locally on the Raspberry Pi, which listens to data posted on a common ROS2 topic. This data is then converted into API commands that the robot can interpret, enabling the robot to move according to its joint state.

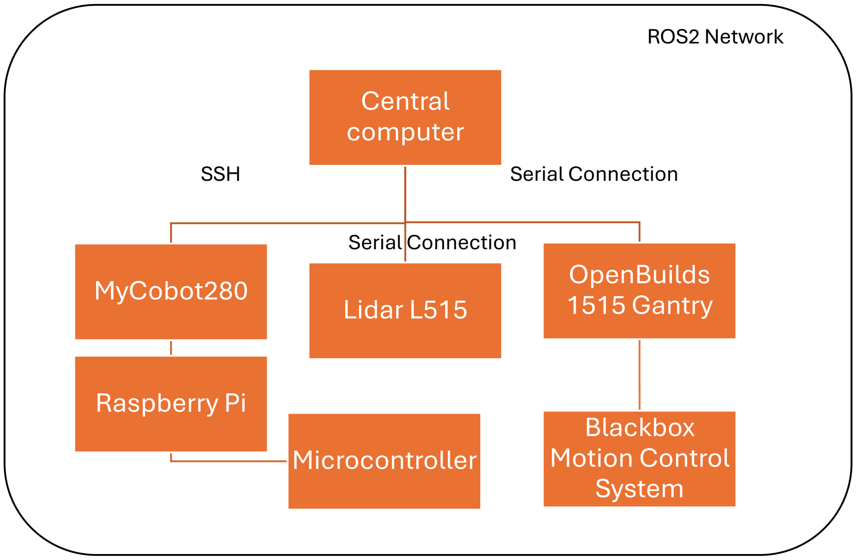


Figure 3.4: Heterogeneous robotic system design featuring the gantry, manipulator, and the LiDAR, along with their connection types.

The Intel RealSense L515 is based on time-of-flight technology, allowing it to capture both RGB and depth images. With the weight of the entire sensor being just 100 grams, it is perfect payload size for the manipulator chosen in this system. The vendor also provides ROS2 packages to integrate these sensor in ROS2 enviroments and the SDK provided can be used to control the sensor through scripts. One limitation of the L515 is that the LiDAR's minimum working distance is 25 centimetres. The L515 provides depth accuracy of up to 9 millimetres at a range of 1 meter, with a maximum range of 9 meters. It offers a horizontal field of view of 70 degrees and a vertical field of view of 43 degrees, providing wide coverage for multi-view imaging. The time-of-flight system works by emitting infrared light pulses and measuring the time it takes for the light to return after reflecting off surfaces, thereby calculating distance. Another advantage of its ROS2 compatibility is that it can be used for real-time data collection and point cloud visualization in tools like RVIZ2, supported by the vendor-provided SDK.

This would make the total degrees of freedom of the entire robot to be 9. An unrestrained

Table 3.2: Joint information of the Mini T-Rex.[27]

Joint Information		
Joint Name	Lower Limit	Upper Limit
joint_x	-1.05m	0m
joint_y	-1.24m	0m
joint_z	-0.1m	0m
joint_1	-165deg	+165deg
joint_2	-165deg	+165deg
joint_3	-165deg	+165deg
joint_4	-165deg	+165deg
joint_5	-165deg	+165deg
joint_6	-175deg	+175deg

rigid body in space have 6 degrees of freedom, 3 for position and 3 for orientation. Theoretically a serial robot with at least 6DoF is needed for the end effector to reach any desired pose of the end effector in its workspace[33]. Having more degrees of freedom than required makes a system kinematically redundant. Redundancy has its own advantages, the end effector can now have infinite number of ways to reach the desired pose. This particularly helps in confined close spaces that might have dynamic elements that can pose as obstacles. It also helps with the convergence time and accuracy of the motion planning algorithms[32, 33]. Singularity condition which can be hard to control are also negated with the additional degrees of freedom. Apart from these advantages, it increases the range of the end effector since the manipulator is attached to a base which can be extend in 3-axis. This makes the robot 3P6R.

3.1.2 Software Description

Collision free execution of the Mini T-Rex depends on a crucial component which, is the digital twin of the robot. This poses a challenge because we need to coordinate between 2 homogeneous robotic systems – the gantry and the manipulator. The novelty of the system is that we are trying to build the system using over the counter products with the firmware provided by the respective vendors, to make the assembly, scaling, and access easier and seamless. Instead of writing combined software for the heterogenous system to behave as a homogenous one, we used the ROS2 framework to control the multi-robot system. To achieve this abstraction of hardware, we combined the two robots in a digital twin.

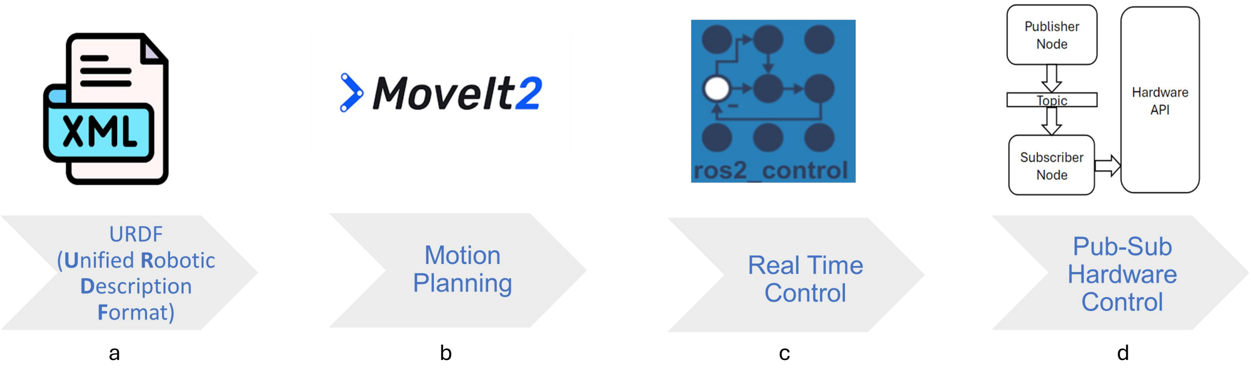


Figure 3.5: Pipeline to control the Mini T-Rex using the digital twin. a) URDF file contains all the static and dynamic properties of the Mini T-Rex, b) Motion planning framework, c) Real time controllers, d) Publisher-Subscriber mode of control.

As shown in Figure 3.5 the first step in the pipeline to control the Mini T-Rex is to design the robots in the URDF (Unified Robot Description Format). To make a good digital replica of the hardware, all the static parameters of both the robots were used to make a CAD design in SolidWorks, using the SW2URDF tool[4] which is a plugin in SolidWorks that converts the CAD design to URDF file. By doing so we can treat this homogenous system at the software level, but use the vendor provided API (Application provided interface) at the hardware level. Since the serial manipulators have revolute joints, the exact static parameters were used as shown in Figure 3.2 while making the URDF file, because not following these would result in a mismatch between the behavior of the digital simulation and the hardware execution. In this case, a change in the direction of the axis of rotation would make the movement inverse of what it should be.

Once the URDF is ready we need to add the configuration files required for this robot to work with MoveIt2 motion planning framework. This involves writing the SRDF (Semantic Robot Description Format) file, the controllers, and some setup configuration files required by the ROS2 framework. We used the MoveIt2 Setup assistant to generate the required files using the URDF file generated. MoveIt2 framework integrates the OMPL (Open Motion Planning Library) which is a collection of various sampling-based motion planning algorithms. By using the motion planner selected, we can then use MoveIt2 either using interactive markers in RVIZ2 which is a visualizer in ROS2 or using either nodes in C++ and python respectively. Figure 3.6 provides a good high-level overview of how the MoveIt2 motion planning framework works and can be used to create collision free path planning.

In ROS2, when a desired position is set by the user, a request is made to the selected motion planner to decide the states of all the joints involved in the simulation. These calculated values are then published to a topic, which is a way to communicate between nodes. Since these are the values of the joints the hardware needs to follow to imitate the digital simulation, subscribers are used for each of the robots which latch on this topic and then convert the received values in API commands. For example, in the case of the manipulator we use the API provided by Elephant Robotics and the gantry is controlled using G-codes. These subscriber nodes extract the values for only the joints that they control from this shared topic, the firmware at both the ends received each command and stores them in queue for execution. This is important to note because this ensures that all the joint states that the planner has calculated are executed to avoid any unplanned motion. By doing so, we synchronize the movement of Mini T-Rex and the digital simulation.

Figure 3.6 provides a good overview of all the different software components used to operate the Mini T-Rex. The user can control the digital twin using either the GUI or write the pose of the end effector in code. The MoveIt2 configuration is loaded using the URDF,

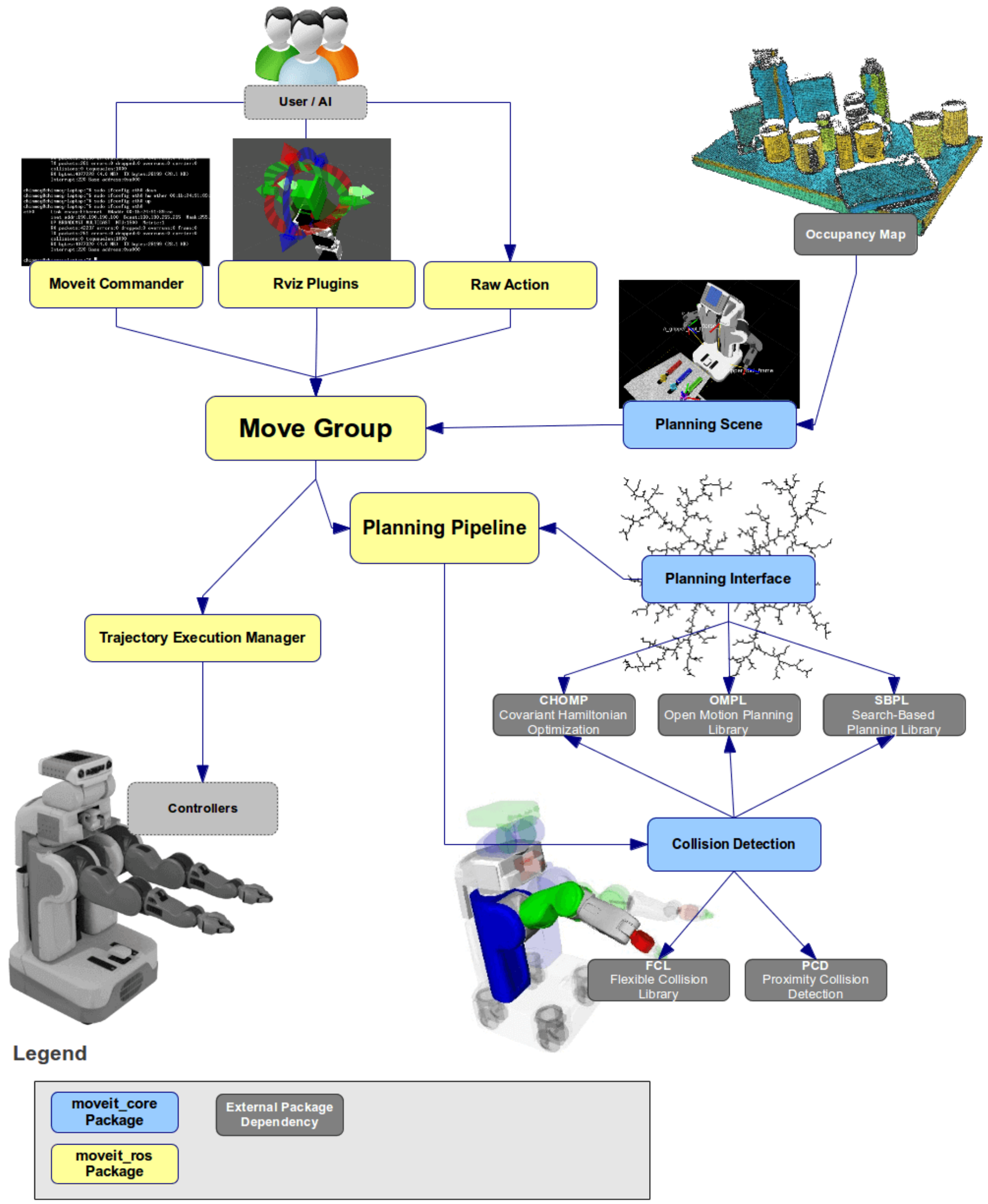


Figure 3.6: MoveIt2 motion planning framework block diagram.Original Diagram from [29]

SRDF and config files where the motion planners will calculate the paths from the start position to the desired end position. Different plugins from the MoveIt2 framework like the collision detection, and the planning interface can interface with the robot interface where the controllers get the commands for the motion, and the robot state publishers take information from the transformation frames, and push the information in the appropriate channels. This information is then used to control the robotic motion.

Figure 3.7 shows the flow of the Mini T-Rex to capture images for one plant. The first step is to start the subscribers for both the robots. When these subscribers started, the first command at every iteration is to let the robot go to the predefined home position. This is done to make sure that, in case of malfunction in previous iteration, the robot can start from a known reference point. In the case of serial manipulator, the joint angles can be reported to the master even if its system hard reboots. In case of the gantry, it does not store the last value after reboots, therefore, the homing action will ensure that the starting point remain the same for all iterations. In the homing action, the gantry has hard stop switches which, when triggered by the contact with the gantry columns, signals the home position of the robot. The gantry used in this work has three such switches for each axis.

The second step is to fix the desired end effector pose. It can be done in various ways, either using interactive markers in RVIZ2, or setting the position and orientation through code. Once this is done, the motion planners make a sequence of joint positions over time as a trajectory of the joints of the robot. Once these joint states are created, the robot matches the joint states to reach the desired pose, after which our sensor captures the RGB and depth images. This process can be repeated for any number of points needed for each plant. For this study, we used 6 pre-defined poses for each plant.

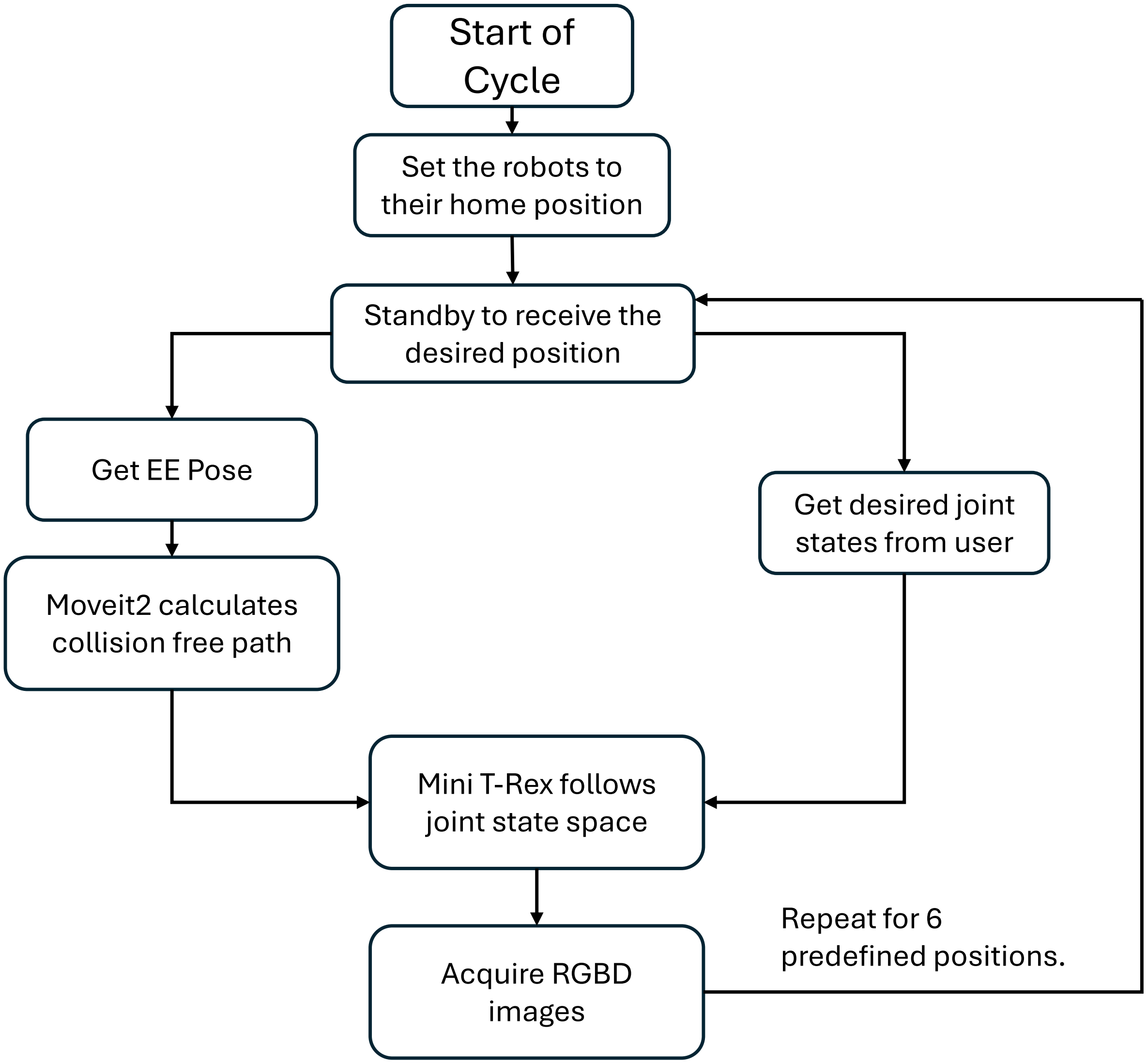


Figure 3.7: Task logic from getting end effector coordinates to capturing RGBD images within one cycle

3.2 Hardware and Software Setup of I2GROW Oasis

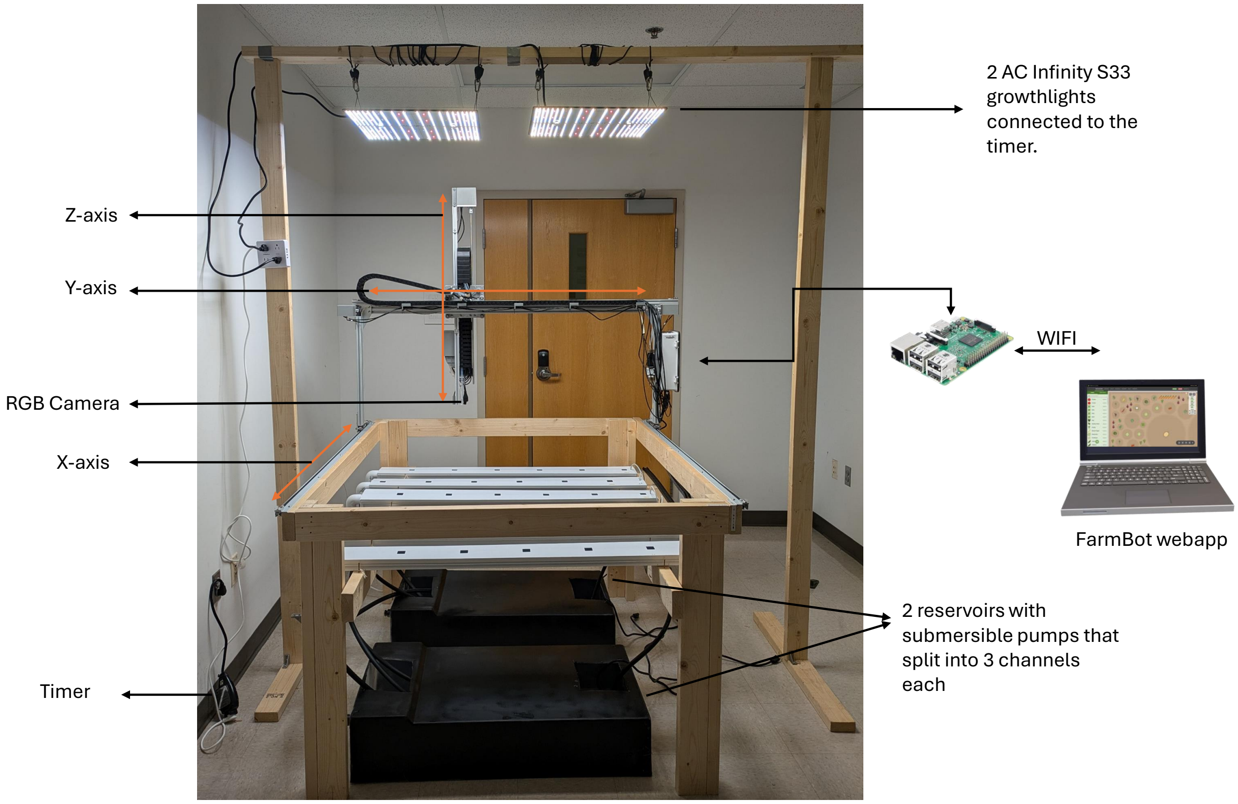


Figure 3.8: FarmBot adapted to the modified NFT hydroponics system with the AC Infinity S33 growth lights.

This section describes another gantry like robot called I2GROW Oasis system, and the specific gantry robot used is called FarmBot.

The FarmBot is available commercially, and can be assembled to monitor and be used for various other agricultural applications. One constraint that the FarmBot has is that the end effector cannot change its orientation since it is a static joint where different sensors can be attached for various applications. Given this constraint, our specific use of FarmBot is to capture images from top-down view, and we only used the borehole camera attached to the FarmBot.

The FarmBot system comes with a web dashboard from where it can be controlled in real time and be used to make schedules and regimes for watering, weeding and seeding. In our case, we capture images of each plant growing in the hydroponic NFT system.

3.2.1 Hardware Description

The FarmBot is mounted on top of a hydroponic setup where the plants that need to be monitored are growing. The setup also has external growth lights, and submersible pumps to feed the plants with nutritional solutions. The lights, controlled by timers, mimic sunlight cycles—set to provide 16 hours of light and their intensity can be adjusted based on the crop species.

The FarmBot functions as a 3-axis linear system, capable of being programmed to reach different coordinates within its working area, capture RGB images, and upload them to the web. The entire outer structure of the FarmBot is 1370mm wide, 1600mm long and 2000mm high, and the working area is 1231mm in x-axis, 1231mm in y-axis, and 252mm in z-axis. It is actuated in the x and y-axis using a belt driven assembly with 2 stepper motors in the x-axis and 1 stepper motor in the y-axis. For the z-axis, a lead screw is used to drive the end effector in the z-axis.

The Farmbot is controlled using the "Farmduino" which is a microcontroller developed by Farmbot, Inc. This is connected to the Raspberry Pi 4B onboard the FarmBot which communicates with the web dashboard to receive instructions or commands for execution.

The hydroponic system was modified to split the system in half, allowing the use of two different nutrient solutions simultaneously. This modification enables the growth of plants aided by different nutrient solutions under the same controlled environmental conditions. Each reservoir has a dedicated submersed pump that maintains a steady flow of nutrients.

The entire robot has 2 sub-systems which were assembled on a custom-made wooden struc-

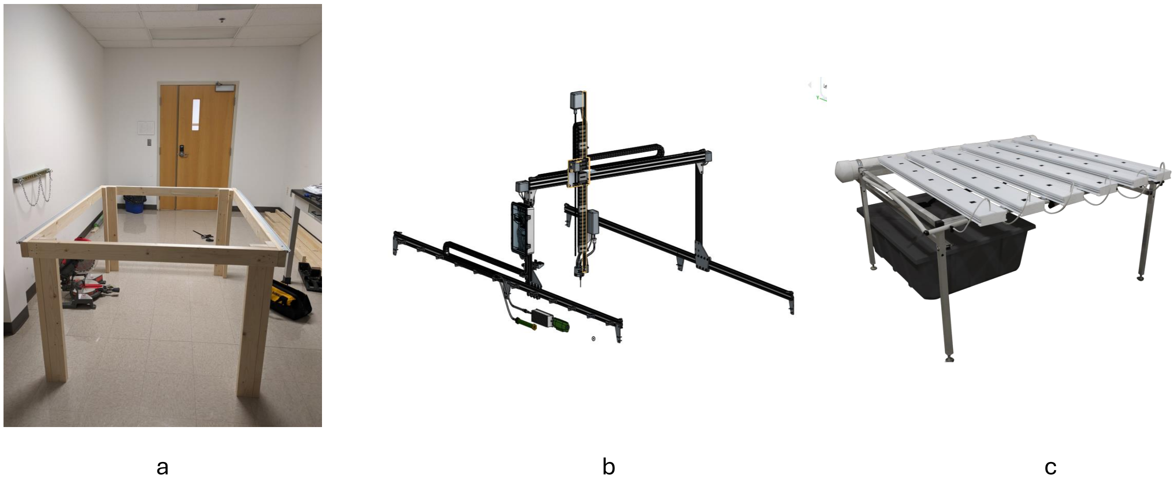


Figure 3.9: Stepwise construction of the I2GROW Oasis adapted for the CropKing NFT hydroponic system. (From left to right) a) Wooden base to support the NFT channels and the FarmBot, b) CAD rendering of the FarmBot, c) CropKing NFT system

ture which accommodates the space required by the hydroponics NFT system and the FarmBot. The version of the FarmBot used in this work is the FarmBot Genesis v1.7 which has dimensions of 3 meters x 1.5 meters. For our test setup, we configured the system to be 1.5m x 1.5m. The CAD rendering of the FarmBot is shown in Figure 3.9 and the conventional way of setting up a hydroponics system in Figure 3.9. In our case, we divided the hydroponic setup into 2 parts with a reservoir each. Alternative channels belong to one tank and these channels are elevated at an angle such that the water will flow in each channel with the help of the submersible pumps and the slope introduced. The submersible pumps are set such that the channels have a flow rate of about 1 liter of solution per minute.

3.2.2 Software Description

The main purpose of the FarmBot in the I2GROW Oasis is to capture images of the plants at predefined positions and intervals. To achieve this, the FarmBot web dashboard provides different ways to configure the FarmBot to set the end effector positions. It can be either done by adding sequences using the widgets on the web dashboard or by creating Lua scripts. Fig 2.10 shows the communication protocols used to send commands to the RPi4B from the web dashboard. The RestAPI is used to store data, and the message broker is used to communicate between the web dashboard and the RPi4B. This requires the FarmBot to be constantly connected to the network over Wi-Fi or ethernet. In our setup a Wi-Fi network was set up for FarmBot. The FarmBot will still work without internet connection, but the tasks that might require storing data or uploading any data to the storage servers will fail.

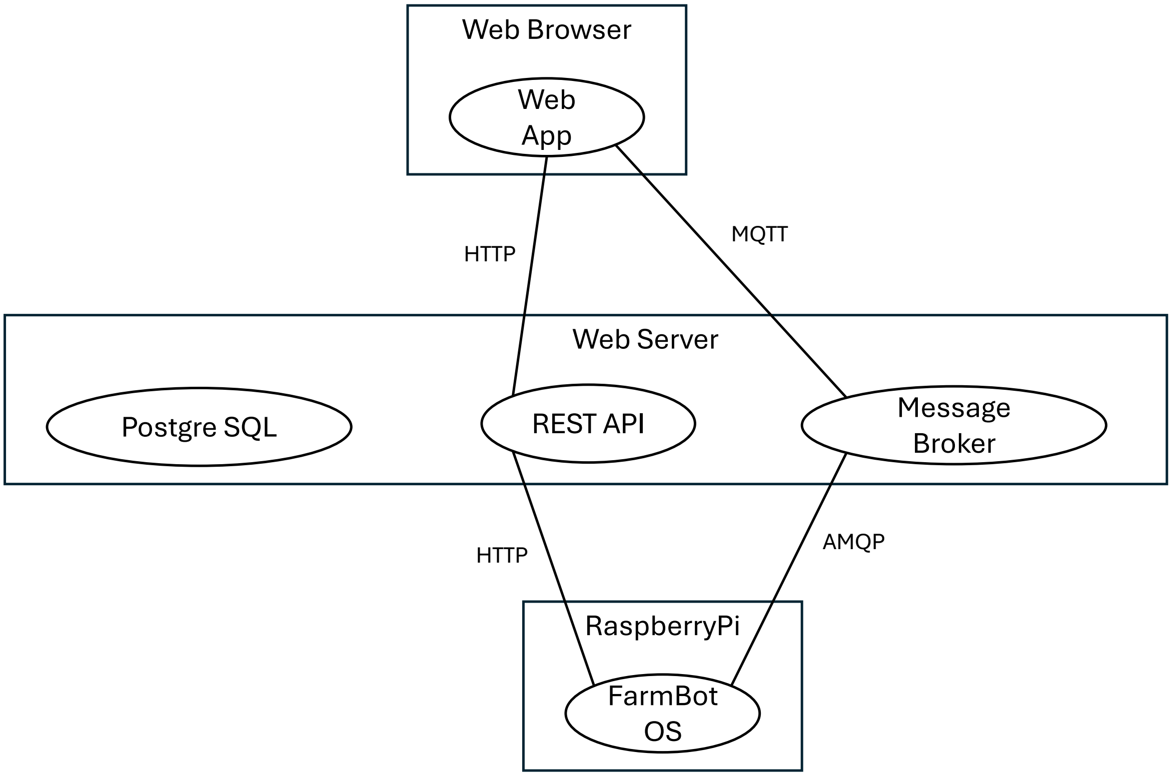
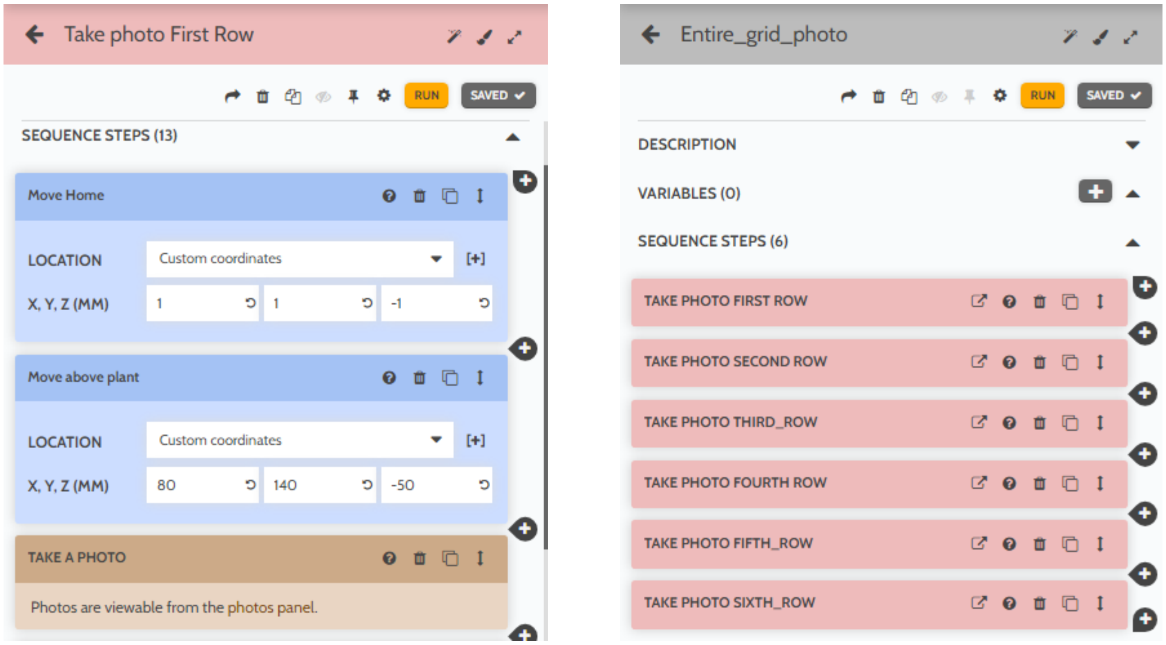


Figure 3.10: Overview of FarmBot communication protocols.

Since there are no moving parts in the hydroponic setup, and the positions of the plants in the channels are fixed throughout the growth cycle, determining positions of these is straightforward. Once the positions are known, sequences can be made as shown in Figure 3.11(a) such that the FarmBot will move above the plant where it needs to take a picture, store the picture, and doing this successively for all the plants. Figure 3.11(b) shows how this can be replicated for all the 6 channels and a regimen can be made which can be repeated for all the intervals where the picture needs to be taken. This lets us program the system such that the end effector reaches the desired position by setting absolute positions and capturing the RGB image after reaching the target. To create longitudinal datasets, we need to run this regimen in regular intervals. The end effector of the FarmBot has a borehole camera attached to it which captures the images. Since the FarmBot is designed on the principles of a precision CNC machine, it provides accuracy in millimeters. Fig 2.12 shows how these sequences can be used to make regimens that run based on the given schedule. One thing to highlight is that the images are saved on the web and not on the local storage of the FarmBot.



Move Home
LOCATION: Custom coordinates
X, Y, Z (MM): 1, 1, -1

Move above plant
LOCATION: Custom coordinates
X, Y, Z (MM): 80, 140, -50

TAKE A PHOTO
Photos are viewable from the photos panel.

a

b

Figure 3.11: Pre-programming the coordinates for the sequence of a) capturing pictures of one row and b) extending a similar sequence for all the channels.

To retrieve these images a bash script was written that extracts the web API keys of the

every_4_hours

VARIABLES (0)

DAY 1

ENTIRE_GRID_PHOTO 6:10am

ENTIRE_GRID_PHOTO 10:10am

ENTIRE_GRID_PHOTO 2:10pm

ENTIRE_GRID_PHOTO 6:10pm

ENTIRE_GRID_PHOTO 9:50pm

DAY 2

ENTIRE_GRID_PHOTO 6:10am

ENTIRE_GRID_PHOTO 10:10am

ENTIRE_GRID_PHOTO 2:10pm

ENTIRE_GRID_PHOTO 6:10pm

ENTIRE_GRID_PHOTO 9:50pm

DAY 3

ENTIRE_GRID_PHOTO 6:10am

ENTIRE_GRID_PHOTO 10:10am

ENTIRE_GRID_PHOTO 2:10pm

Figure 3.12: Setting up a regime to take pictures using the sequences depicted in Fig2.11 in intervals of 4 hours everyday.

images and then download them individually. These keys are stored chronologically, and the bash script saves them in that order with the last taken image is downloaded first. The bash script runs to download the last 36 images which match the number of plants growing. The bash script was written for Linux and has an equivalent file in PowerShell for a windows system. Whenever this script is executed, it will download the last 36 captured images and store them with chronological file names in time dated folders.

Since the time when the regimen runs are fixed as shown in Fig 2.12, the bash script is used to set up a CRON job that runs this bash script at an offset of 1 hour after every capture cycle.

Chapter 4

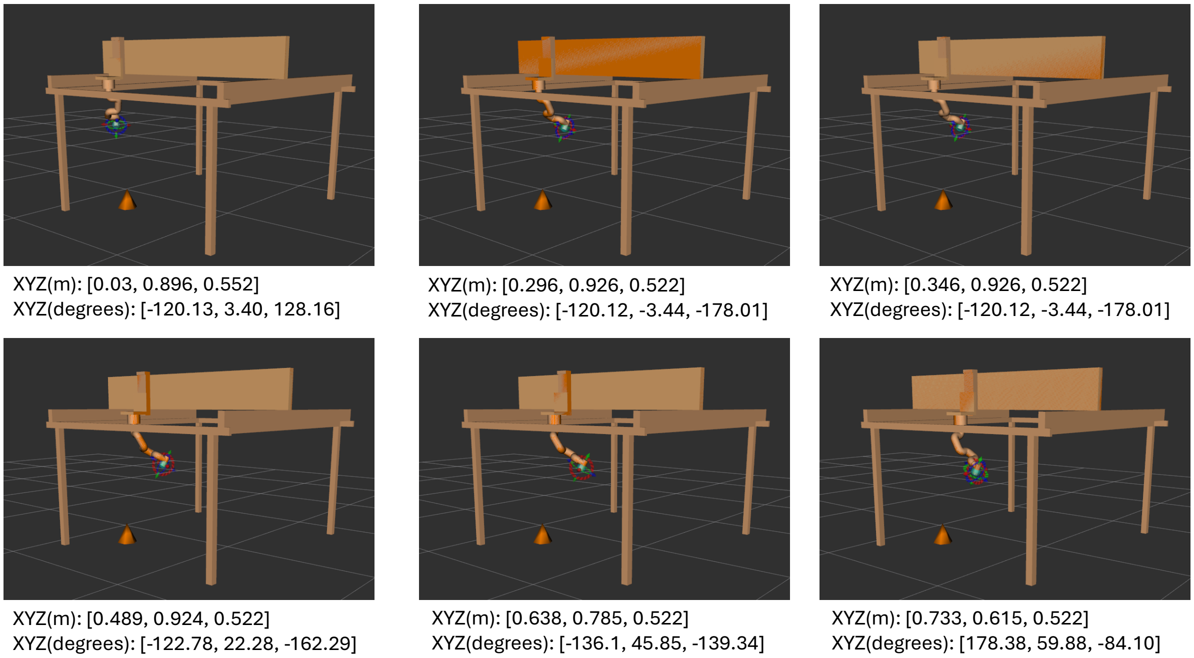
Results

In this section, we will present the data captured by the robotic systems, discuss how the data was captured, and compare the advantages of each design when it comes to collecting images through an automated pipeline. The results section is divided in 2 parts, one for the Mini T-Rex and one for the I2GROW Oasis. The primary goals for both robotic systems are to gather phenotypic data. For the Mini T-Rex, we intend to show a proof of concept that the Mini T-Rex is capable of multi-view image capture for 3D-reconstruction, and assessment of the accuracy with which the robotic system reaches the coordinates on every iteration. The I2GROW Oasis focuses more on the reliable schedulability and the image captures to create longitudinal datasets from accurate predefined points repeatedly.

4.1 Mini T-Rex

The Mini T-Rex goes to 6 predefined positions as shown in Figure 4.1 to capture the RGBD data from different orientations of the same plant. In Figure 4.1, the pose shown are exactly the same which the Mini T-Rex goes to to capture the RGBD data. The images collected to show the pose of the robot is collected from RVIZ2 where the orange cone can be assumed as the location of the test plant in real life and the pose of the robot can be set accordingly.

Growth Conditions of Test Plants The plants used to test this system were grown away from the Mini T-Rex growth bed due to space constraints. The ideal scenario is



XYZ(m): [0.03, 0.896, 0.552]
XYZ(degrees): [-120.13, 3.40, 128.16]

XYZ(m): [0.296, 0.926, 0.522]
XYZ(degrees): [-120.12, -3.44, -178.01]

XYZ(m): [0.346, 0.926, 0.522]
XYZ(degrees): [-120.12, -3.44, -178.01]

XYZ(m): [0.489, 0.924, 0.522]
XYZ(degrees): [-122.78, 22.28, -162.29]

XYZ(m): [0.638, 0.785, 0.522]
XYZ(degrees): [-136.1, 45.85, -139.34]

XYZ(m): [0.733, 0.615, 0.522]
XYZ(degrees): [178.38, 59.88, -84.10]

Figure 4.1: Visual representation of the 6 predefined poses of the Mini T-Rex with the cartesian position in meters and the Euler angles in degrees at the bottom of each pose.

growing the plants under the Mini T-Rex on grow beds with the help of growth lights which would showcase the sensor to plant capabilities of Mini T-Rex. The plants chosen for this experiment were the Solanum lycopersicum ’Early Girl’ and Thlaspi arvense. The photoperiod was kept at 16 hours, and the plants were watered three times a week without any fertilizers.

One of the constraints for this project was the plants were not growing under the gantry which is the ideal condition. Due to space constraints, our test plants which include pennycress and Early girl hybrid tomato plants were grown under controlled conditions away from the Mini T-Rex grow bed. Each time the plant needs to be sampled, it was kept under the Mini T-Rex while maintaining proper orientation of the plant to enable data alignment, since the data gathered was for a span of 30 days. To do so, a location was marked on the test bed and each test plant had a fixed orientation when it was kept on the fixed location on the test bed. This was done to ensure that there was consistency in the plant positioning, to showcase the precise positioning of the sensor.

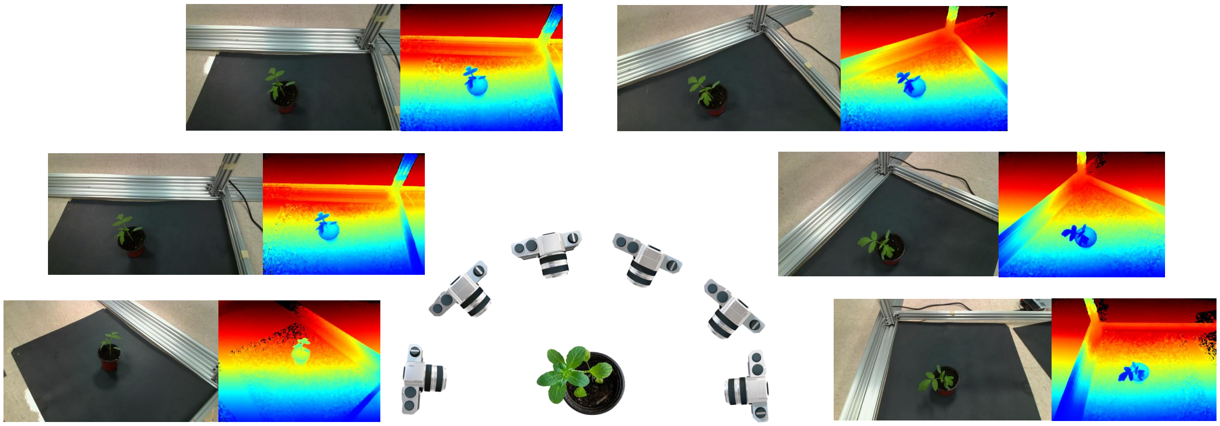


Figure 4.2: Resultant RGB and depth images taken of an early girl tomato plant from the 6 poses illustrated in Fig 3.1

Figure 4.2 shows a rough orientation of the camera with respect to the plant when the sensor captures the data. The plant is oriented such that it shows the top view. Each pose has a corresponding collage of the resultant RGB and depth images taken which demonstrates the different point of views from where the plant is being sampled which will ultimately be useful when 3D reconstructing the phenotype. Each collage has the RGB image and corresponding depth image on the right. The plants were sampled three times a week in this period since the size of the plants was significant. Figure 4.3 shows the progression in growth of the plant over a period of 30 days.

4.2 I2GROW Oasis

The grow bed of the I2GROW Oasis is the hydroponic systems which has a total of 6 channels with 6 holes each for plants to grow, making a total of 36 plants that can grow in one growth cycle. The Figure 4.4 shows how these slots are placed with respect to the rails

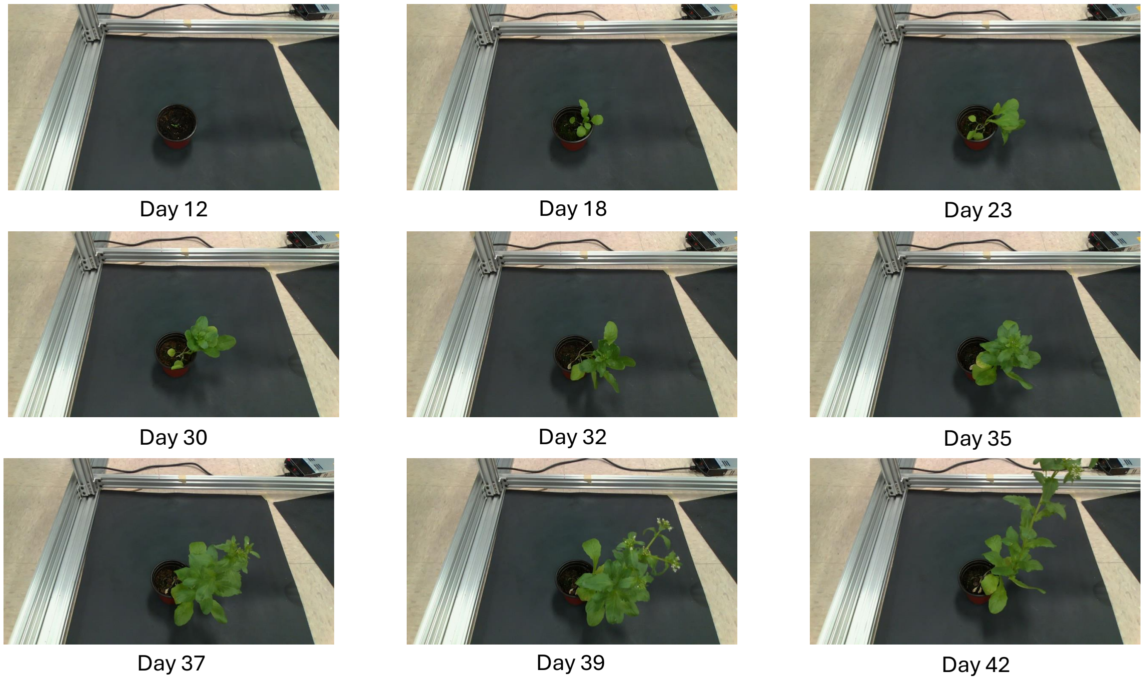

Day 12
Day 18
Day 23
Day 30
Day 32
Day 35
Day 37
Day 39
Day 42

Figure 4.3: Image of the same pennycress plant from Pose 6(Figure 3.1) taken every alternate day.

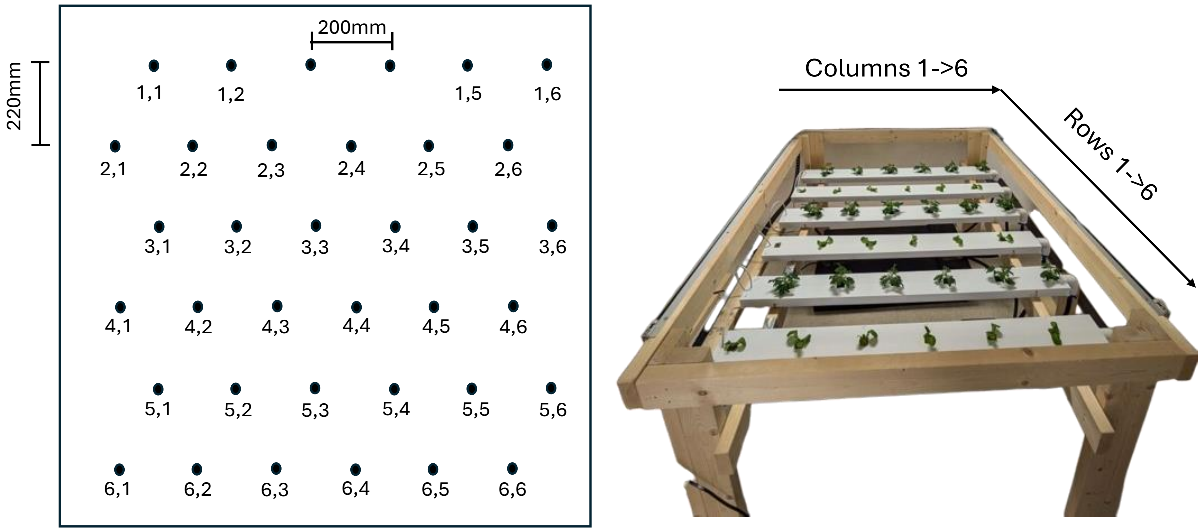


Figure 4.4: Coordinates of the camera position in the 2d plane and the way the channels are numbered.

of the FarmBot. The channels 1, 3, and 5 were attached to one reservoir and the channels 2, 4, and 6 were connected to another reservoir.

In section 3.2.2 we discussed how the bash script is run like a CRON job, which downloads the images taken from the RGB camera to a local system where the bash script is running and the images are stored in this format:

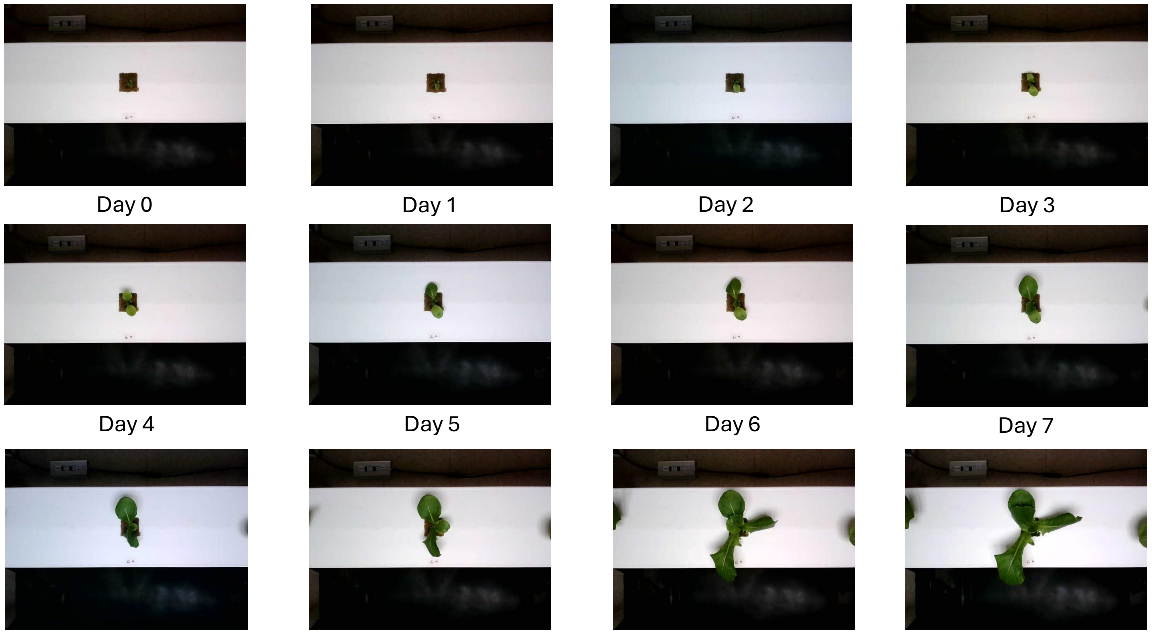
farmbot_images/YYYYMMDD_HHMMSS/image_n.png

The parent folder is where the bash script is saved, each iteration makes a folder with the name as a combination of the current date and time. The 'n' in the image name is number of the grid point as shown in 4.4 as an integer. The way that the API keys are stored on the FarmBot web-app is 'Last in First Out' such that the latest image is obtained first and hence the numbering starts with that, that is coordinate 6,6 will have name as "image_1.png" and 1,1 will have name as "image_36.png".

Controlled Environment Growth Conditions For this growth cycle the 6 channels were divided equally between Solanum lycopersicum 'Micro-Tom' and Parris Island Cos Romaine Lettuce. These alternate channels receive nutrient solutions through dedicated tanks, that have a solution of 'Jack's Nutrients' Part A and Part B mized as per the instruction provided by the manufacturer ie. 4 grams per gallon for Type A and 2.5 grams per gallon for Type B. The plants are first sown in rock wool, germinated and then transferred into the respective slots in the hydroponic system.

As shown in Figure 4.5 each slot is offset by the previous one by a fixed distance of 200mm in each row and the channels are spaced out such that the perpendicular distance between the two points is 220mm. The camera of the FarmBot follows the grid in the same order. The camera was scheduled to take pictures everyday in 4 hour intervals.

The Figure 4.5 and Figure 4.6 show the accuracy with which the FarmBot can take pictures



Day 0 Day 1 Day 2 Day 3

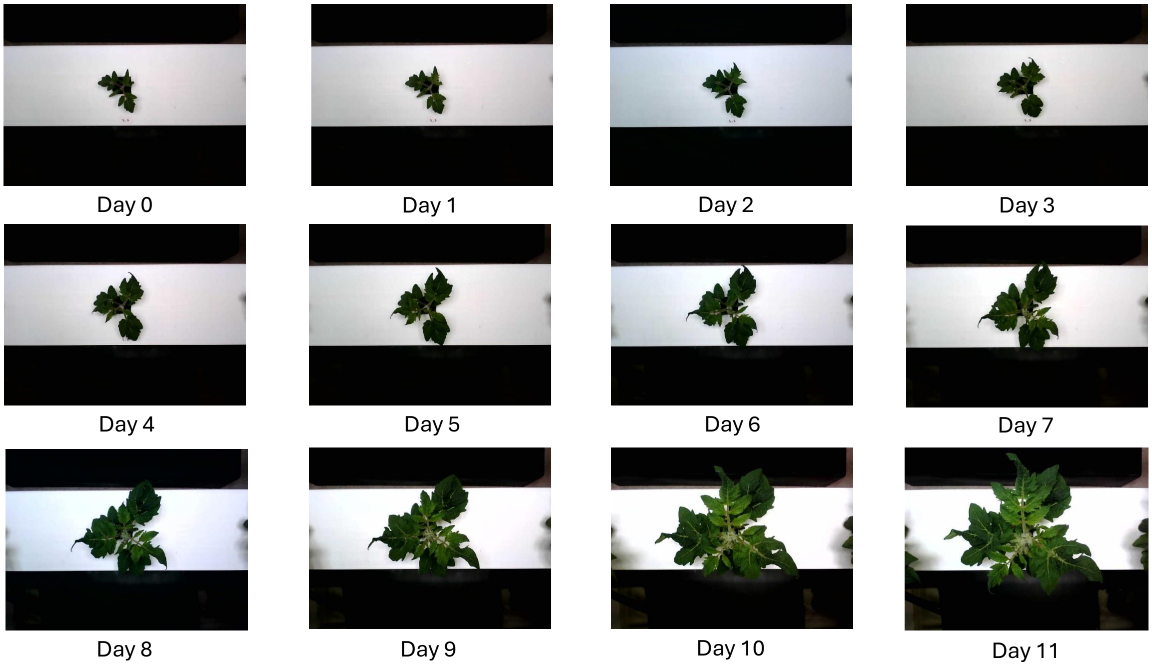
Day 4 Day 5 Day 6 Day 7

Day 8 Day 9 Day 10 Day 11

Figure 4.5: Serial images of the same lettuce plant over 10 days.

from the same location over iterations. By compiling such images, it can be helpful for other applications like leaf tracking and canopy tracking.

The images shown are of every alternate day so that visible difference in the plant growth cycle can be seen by the human eye. With the FarmBot taking pictures every 4 hours it can help in evaluating patterns, change in growth environment and the effects of it, monitoring and tracking growth. Close sampling times can also help in creating time-lapses.



Day 0 Day 1 Day 2 Day 3

Day 4 Day 5 Day 6 Day 7

Day 8 Day 9 Day 10 Day 11

Figure 4.6: Serial images of the same micro-tom plant over 10 days.

Chapter 5

Conclusion and Future Work

In conclusion, we present Mini T-Rex, a ROS2 compatible robotic system composed of 2 robots which are available for purchase over the counter and can be easily combined to replicate our setup. The system runs on the code developed in this project, and can easily be scaled for other applications. We demonstrate the pipeline for any user to follow. This pipeline started from a CAD design of the robot, and scripts to control hardware. We hope this pipeline can be followed by plant researchers with minimal robotics expertise to build similar robots. The Mini T-Rex is capable of going to any desired location in the workspace while avoiding obstacles and planning paths without collisions. While our aim is to demonstrate a general robotic base, we show one application in the form of collecting phenotypic data by collecting multi-view images from different perspectives of the plant and using L515 sensor to capture RGBD images.

In addition to the mini T-Rex, we show another gantry like robot known as FarmBot and demonstrate how it can be combined with plant growth systems to automate the process of capturing longitudinal image data, and saving it to the local system to document plant growth. While doing the experiments to collect the data, we collected more than 100 sets of data of micro tomato and pennycress plants similar to the ones mentioned in figure 4.5 and 4.6.

Future work can include making this robotic system a closed-loop system where the feedback from both the robots can be used to update the joint positions in software to adapt to dynamic changes, or to make the system safer in case of system freezing. By doing so, the L515 sensor can be used to map in real time the dynamic environment which will be one step towards making the entire system fully autonomous.

Another future direction can be to design different end effector attachments like micro-needles for DNA extraction, or water spouts for irrigation. These end effectors should be easily attached and detached, which can make the entire system self sufficient, as inspired by the end effector designs of the FarmBot.

As for the I2GROW Oasis systems, different sensors that can monitor the CO2 content, EC and pH levels, and humidity. These environmental factors can then be used to maintain the desired levels autonomously without any human intervention.

Bibliography

[1] Abbas Atefi, Yufeng Ge, Santosh Pitla, and James Schnable. Robotic technologies for high-throughput plant phenotyping: Contemporary reviews and future perspectives. *Frontiers in plant science*, 12:611940, 2021.

[2] Andrea Bonci, Francesco Gaudeni, Maria Cristina Giannini, and Sauro Longhi. Robot operating system 2 (ros2)-based frameworks for increasing robot autonomy: A survey. *applied sciences*, 13(23):12796, 2023.

[3] Andrea Botta, Paride Cavallone, Lorenzo Baglieri, Giovanni Colucci, Luigi Tagliavini, and Giuseppe Quaglia. A review of robots, perception, and tasks in precision agriculture. *applied mechanics*, 3(3):830–854, 2022.

[4] Stephen Brawner. Export a solidworks assembly to urdf. `https://wiki.ros.org/sw_urdf_exporter/Tutorials/Export%20an%20Assembly`.

[5] Ayan Chaudhury, Christopher Ward, Ali Talasaz, Alexander G. Ivanov, Mark Brophy, Bernard Grodzinski, Norman P. A. Hüner, Rajnikant V. Patel, and John L. Barron. Machine vision system for 3d plant phenotyping. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 16(6):2009–2022, 2019. doi: 10.1109/TCBB.2018.2824814.

[6] Ziling Chen, Jialei Wang, and Jian Jin. Fully automated proximal hyperspectral imaging system for high-resolution and high-quality in vivo soybean phenotyping. *Precision Agriculture*, 24(6):2395–2415, 2023.

[7] Sachin Chitta, Ioan Sucan, and Steve Cousins. Moveit![ros topics]. *IEEE robotics & automation magazine*, 19(1):18–19, 2012.

[8] Hong-Beom Choi, Jae-Kun Park, Soo Hyun Park, and Taek Sung Lee. Nerf-based 3d reconstruction pipeline for acquisition and analysis of tomato crop morphology. *Frontiers in Plant Science*, 15:1439086, 2024.

[9] Kshitiz Dhakal, Upasana Sivaramakrishnan, Xuemei Zhang, Kassaye Belay, Joseph Oakes, Xing Wei, and Song Li. Machine learning analysis of hyperspectral images of damaged wheat kernels. *Sensors*, 23(7):3523, 2023.

[10] Harnaik Dhami, Kevin Yu, Tianshu Xu, Qian Zhu, Kshitiz Dhakal, James Friel, Song Li, and Pratap Tokekar. Crop height and plot estimation for phenotyping from unmanned aerial vehicles using 3d lidar. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2643–2649. IEEE, 2020.

[11] Tom Duckett, Simon Pearson, Simon Blackmore, Bruce Grieve, Wen-Hua Chen, Grzegorz Cielniak, Jason Cleaversmith, Jian Dai, Steve Davis, Charles Fox, et al. Agricultural robotics: the future of robotic agriculture. *arXiv preprint arXiv:1806.06762*, 2018.

[12] Jonathan D Gammell, Timothy D Barfoot, and Siddhartha S Srinivasa. Batch informed trees (bit*): Informed asymptotically optimal anytime search. *The International Journal of Robotics Research*, 39(5):543–567, 2020.

[13] Ioannis Gravalos, Avgoustinos Avgousti, Theodoros Gialamas, Nikolaos Alfieris, and Georgios Paschalidis. A robotic irrigation system for urban gardening and agriculture. *Journal of agricultural engineering*, 50(4):198–207, 2019.

[14] Intel. Intel realsense lidar l515. https://www.intelrealsense.com/lidar-camera-l515/.

[15] Jawad Iqbal, Rui Xu, Hunter Halloran, and Changying Li. Development of a multipurpose autonomous differential drive mobile robot for plant phenotyping and soil sensing. *Electronics*, 9(9):1550, 2020.

[16] Lydia E. Kavraki and Steven M. LaValle. *Motion Planning*, pages 139–162. Springer International Publishing, Cham, 2016. ISBN 978-3-319-32552-1. doi: 10.1007/978-3-319-32552-1_7. URL https://doi.org/10.1007/978-3-319-32552-1_7.

[17] Steven M La Valle. Motion planning. *IEEE Robotics & Automation Magazine*, 18(2):108–118, 2011.

[18] Kantor Lab. Robot for extracting dna samples from tomato plants. https://labs.ri.cmu.edu/kantorlab/tomato-gantry/.

[19] David LeBauer, Max Burnette, Noah Fahlgren, Rob Kooper, Kenton McHenry, and Abby Stylianou. What does terra-ref's high resolution, multi sensor plant sensing public domain data offer the computer vision community? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1409–1415, 2021.

[20] Song Li. Caff i2grow: Intelligent and integrated greenhouse gas removal with indoor and controlled environment agriculture. https://portal.nifa.usda.gov/web/crisprojectpages/1030119-caff-i2grow-intelligent-and-integrated-\greenhouse-gas-removal-with-indoor-and-controlled-environment-agriculture.html.

[21] Francisco Lourenço and Helder Araujo. Intel realsense sr305, d415 and l515: Experi-

mental evaluation and comparison of depth estimation. In *VISIGRAPP (4: VISAPP)*, pages 362–369, 2021.

[22] Mohd Saiful Azimi Mahmud, Mohamad Shukri Zainal Abidin, Abioye Abiodun Emmanuel, and Hameedah Sahib Hasan. Robotics and automation in agriculture: present and future applications. *Applications of Modelling and Simulation*, 4:130–140, 2020.

[23] Theo Moons, Luc Van Gool, Maarten Vergauwen, et al. 3d reconstruction from multiple images part 1: Principles. *Foundations and Trends® in Computer Graphics and Vision*, 4(4):287–404, 2010.

[24] Fumio Okura. 3d modeling and reconstruction of plants and trees: A cross-cutting review across computer graphics, vision, and plant phenotyping. *Breeding Science*, 72 (1):31–47, 2022.

[25] Luiz FP Oliveira, António P Moreira, and Manuel F Silva. Advances in agriculture robotics: A state-of-the-art review and challenges ahead. *Robotics*, 10(2):52, 2021.

[26] OpenBuilds. Openbuilds lead 1515 website. `https://us.openbuilds.com/lead-cnc-machine-1515-60-x-60/`.

[27] Elephant Robotics. Gitbook for mycobot280pi. `https://docs.elephantrobotics.com/docs/gitbook-en/2-serialproduct/2.1-280/2.1-280.html`, .

[28] Elephant Robotics. Mycobot280pi manual. `https://docs.elephantrobotics.com/docs/gitbook-en/2-serialproduct/2.1-280/2.1.5.1-IntroductionOfProductParameters.html`, .

[29] PickNik Robotics. Moveit2 block diagram. `https://moveit.picknik.ai/main/doc/concepts/concepts.html`, .

[30] Arno Ruckelshausen, Peter Biber, Michael Dorna, Holger Gremmes, Ralph Klose, Andreas Linz, R Rahe, Rainer Resch, Marius Thiel, Dieter Trautz, et al. Bonirob: an autonomous field robot platform for individual plant phenotyping. In *Precision agriculture'09*, pages 841–847. Wageningen Academic, 2009.

[31] F. Sherwani, Muhammad Mujtaba Asad, and B.S.K.K. Ibrahim. Collaborative robots and industrial revolution 4.0 (ir 4.0). In *2020 International Conference on Emerging Trends in Smart Technologies (ICETST)*, pages 1–5, 2020. doi: 10.1109/ICETST49965.2020.9080724.

[32] Bruno Siciliano. Kinematic control of redundant robot manipulators: A tutorial. *Journal of intelligent and robotic systems*, 3:201–212, 1990.

[33] Abhisesh Silwal, Francisco Yandun, Anjana K Nellithimaru, Terry Bates, and George Kantor. Bumblebee: A path towards fully autonomous robotic vine pruning. *Field Robotics*, 2(1):1661–1696, 2022.

[34] Abhisesh Silwal, Xuemei M Zhang, Thomas Hadlock, Jacob Neice, Shadab Haque, Adwait Kaundanya, Chang Lu, Boris A Vinatzer, George Kantor, and Song Li. Towards an ai-driven cyber-physical system for closed-loop control of plant diseases. In *Proceedings of the AAAI Symposium Series*, volume 4, pages 432–435, 2024.

[35] Howard M. Sparrow R. Robots in agriculture: prospects, impacts, ethics, and policy. *Precision Agric*, (22):818–833, 2021. doi: https://doi.org/10.1007/s11119-020-09757-9.

[36] Grant Takara, A. Zachary Trimble, Reika Arata, Shane Brown, Hector Jaime Gonzalez, and Camilo Mora. An inexpensive robotic gantry to screen and control soil moisture for plant experiments. *HardwareX*, 9:e00174, 2021. ISSN 2468-0672. doi: https://doi.org/10.1016/j.ohx.2021.e00174. URL `https://www.sciencedirect.com/science/article/pii/S2468067221000031`.

[37] Tang Tao, Longfei Gao, Guangrun Wang, Yixing Lao, Peng Chen, Hengshuang Zhao, Dayang Hao, Xiaodan Liang, Mathieu Salzmann, and Kaicheng Yu. Lidar-nerf: Novel lidar view synthesis via neural radiance fields. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 390–398, 2024.

[38] Kaihua Wei, Shuang Liu, Qingguang Chen, Shentao Huang, Mingwei Zhong, Jingcheng Zhang, Hongwei Sun, Kaihua Wu, Shanhui Fan, Ziran Ye, et al. Fast multi-view 3d reconstruction of seedlings based on automatic viewpoint planning. *Computers and Electronics in Agriculture*, 218:108708, 2024.

[39] Xing Wei, Marcela A Johnson, David B Langston Jr, Hillary L Mehl, and Song Li. Identifying optimal wavelengths as disease signatures using hyperspectral sensor and machine learning. *Remote Sensing*, 13(14):2833, 2021.

[40] Jingwen Wu, Xinyu Xue, Songchao Zhang, Weicai Qin, Chen Chen, and Tao Sun. Plant 3d reconstruction based on lidar and multi-view sequence images. *International Journal of Precision Agricultural Aviation*, 1(1), 2018.

[41] Lirong Xiang, Trevor M Nolan, Yin Bao, Mitch Elmore, Taylor Tuel, Jingyao Gai, Dylan Shah, Ping Wang, Nicole M Huser, Ashley M Hurd, et al. Robotic assay for drought (road): An automated phenotyping system for brassinosteroid and drought responses. *The Plant Journal*, 107(6):1837–1853, 2021.

[42] Zhang Yanwen, Hu Kai, and Wang Pengsheng. Review of 3d reconstruction algorithms. *Nanjing Xinxi Gongcheng Daxue Xuebao*, 12(5):591–602, 2020.

[43] Dajun Yu, Nick Lord, Justin Polk, Kshitiz Dhakal, Song Li, Yun Yin, Susan E Duncan, Hengjian Wang, Bo Zhang, and Haibo Huang. Physical and chemical properties of edamame during bean development and application of spectroscopy-based machine learning methods to predict optimal harvest time. *Food Chemistry*, 368:130799, 2022.

[44] Shuwan Yu, Xiaoang Liu, Qianqiu Tan, Zitong Wang, and Baohua Zhang. Sensors, systems and algorithms of 3d reconstruction for smart agriculture and precision farming: A review. *Computers and Electronics in Agriculture*, 224:109229, 2024.

[45] Xuemei Zhang, Boris A Vinatzer, and Song Li. Hyperspectral imaging analysis for early detection of tomato bacterial leaf spot disease. *Scientific Reports*, 14(1):27666, 2024.

Appendices

Appendix A

Image Embeddings

The Figure A.1 shows the image embeddings of all the images captured from the Mini T-Rex. The ResNet50 model was used to determine the image embeddings and t-SNE (t-Distributed Stochastic Neighbor Embedding) was used to lower the dimension of the embeddings to visualize. In Figure A.1 different colors show the pose from which the image was taken, since this already known. Clustering of the markers of the same color shows that the embedding or the features of the plants taken from the same pose are similar which also shows consistency in the positioning of the Mini T-Rex end effector and the formation of different clusters signifies the different features being extracted from the images. The Figure A.2 has the actual images as the thumbnails in the scatter plot, with a closer look we can see the orientation of the camera for all these images in the clusters is the same.

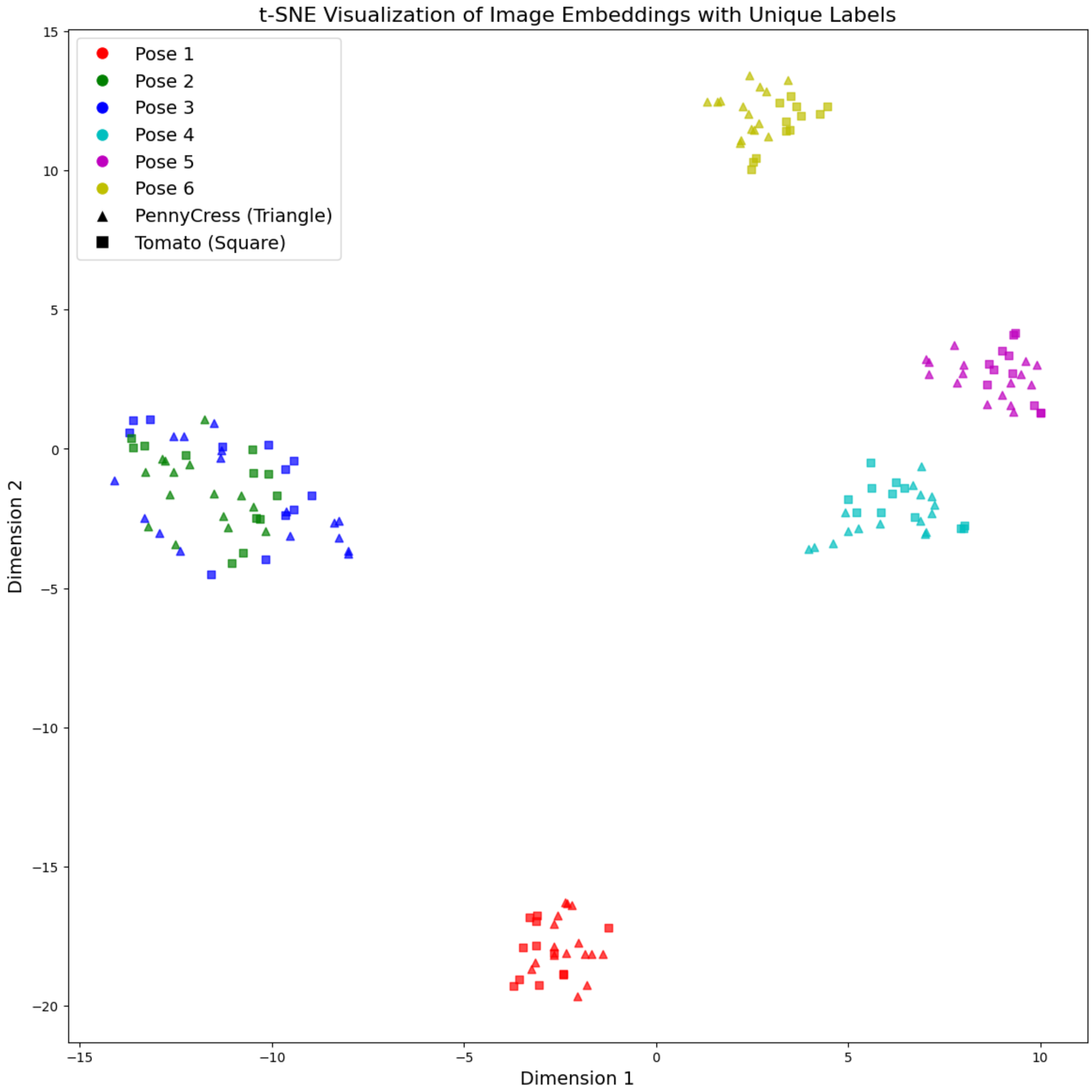


Figure A.1: Image embeddings of images captured on Mini T-Rex for tomato and pennycress plants.

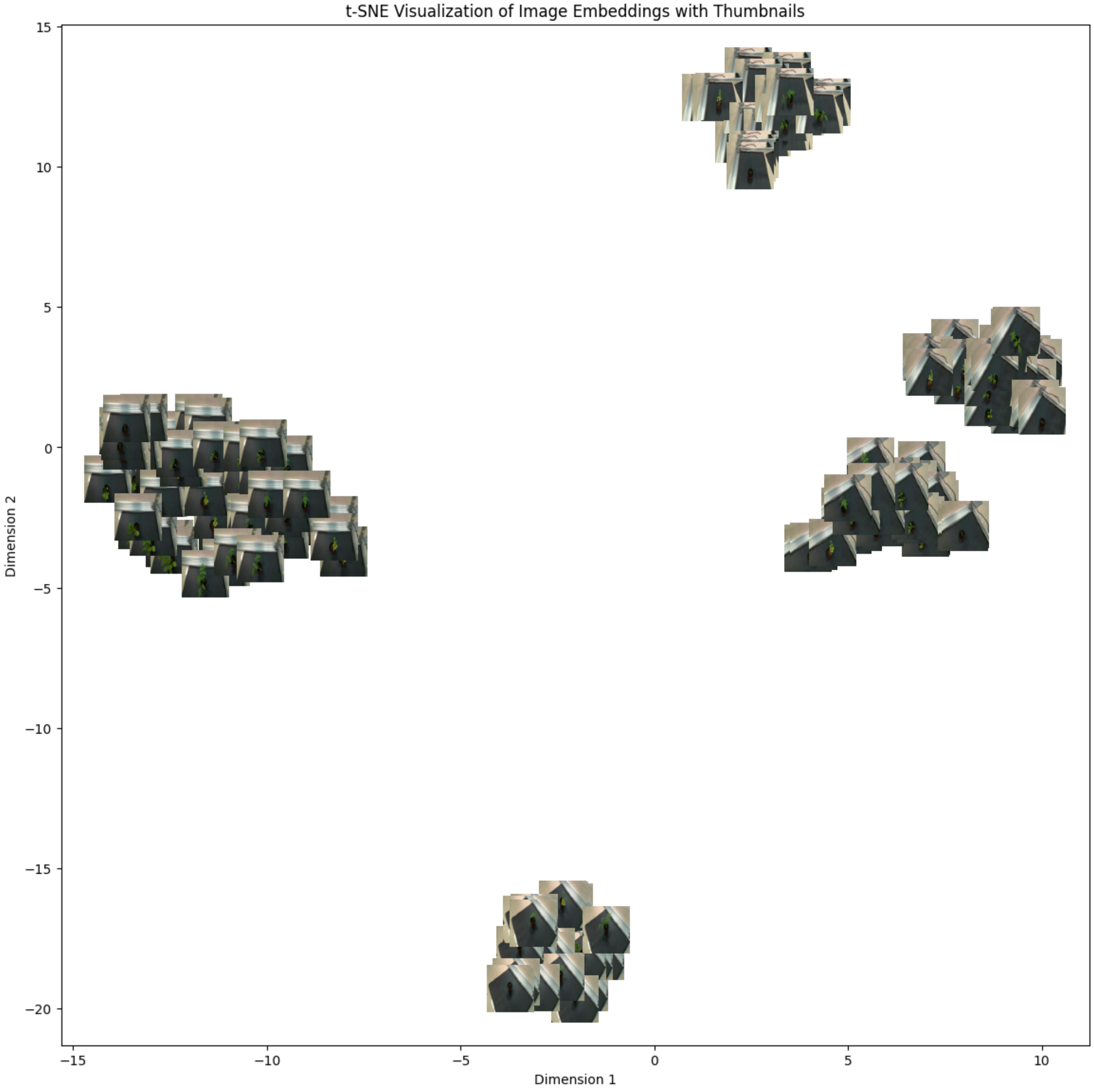


Figure A.2: Scatter plot of image embedding with the respective images as thumbnails.