

Design in Mechatronics and Robotics II, MRE481

Final Report

Project Name: Precision Spraying Robot

Group 3: Jiayi Yuan, Yuchen Xia, Qiuhan Ma, Haibo Zhao

Date: 4/24/2024

Contents

Abstract	5
1 Introduction and Problem Statement.....	7
1.1 Introduction	7
1.2 Problem Statement.....	7
1.3 Preliminary Design Goals.....	8
2 Background Research and Literature Review	9
2.1 Detail Background and Market Research	9
2.2 Customer Needs.....	10
3 Design	10
3.1 Preliminary Design Ideas.....	10
3.2 Design trade study	13
3.3 Target/Final Specifications	14
3.4 Product Architecture.....	15
3.5 Discussion of Constraints, Risks and Back-up Plans*	17
4 Analysis	18
4.1 Analysis Plans.....	18
4.2 Robotic Arm Analysis	19
4.3 Camera and Image Recognition Analysis.....	29

4.3.1 Model Performance Evaluation.....	29
4.3.2 Distance estimation	31
4.3.3 Test Results	33
4.4 Spray control system analysis.....	35
5 Alpha Prototype	44
5.1 Alpha Prototype Construction	44
5.2 Alpha Prototype Testing	45
5.3 Alpha prototype build status.....	45
5.4 Experiment design and testing plan status for Alpha prototype	46
5.4.1 Experiment design.....	46
5.4.2 Testing plan status.....	47
5.5 Test Results.....	47
5.5.1 Spray system	47
5.5.2 Robotic arm.....	49
6 Beta Prototype.....	52
6.1 Beta Prototype Construction.....	52
6.2 Beta Prototype Testing	53
6.3 Beta prototype build status	54
6.4 Experiment design and testing plan status for beta prototype	55
6.4.1 Experiment design.....	55

6.4.2 Testing plan status.....	56
6.5 Test results.....	57
6.5.1 Spray system	57
6.5.2 Camera	62
6.5.3 Robotic arm.....	71
7 Final Product Design.....	79
8 Miscellaneous	81
8.1 Cost Analysis	81
8.2 Timeline.....	82
8.3 Learning Plans and New Knowledges Acquired	83
8.4 Codes and Standards.....	84
8.5 Ethical Issues and Societal Impacts.....	87
9 Next Steps.....	89
10 Conclusion.....	90
References	92
Appendices	94

Abstract

The primary problem this project addresses is the inefficient and excessive application of pesticides in agriculture. Traditional pesticides application methods often lead to pesticides drifting beyond the intended targets and applying a uniform amount of chemicals across fields without considering the specific needs of each crop or plant. This not only wastes resources but also contributes to environmental pollution and increases the risk of harming non-target organisms.

The importance of solving this problem lies in the need to optimize agricultural practices to support sustainable farming. By reducing the volume of pesticides used, it is possible to decrease production costs and minimize ecological footprints. Furthermore, precision spraying can help in maintaining crop health more effectively by targeting only those areas that need treatment, thereby ensuring better crop yield and quality.

This project is designed to effectively control crop lesions while minimizing pesticide waste and reducing environmental impact. It seeks to automate the pesticide application process, enhancing its efficiency using advanced technologies such as image recognition, with the ultimate objective of developing a commercially viable product for agricultural use.

Our project has successfully achieved all its intended functions, so we think it is successful as a senior design project. However, there are areas for future development to enhance its functionality and transition it into a market-ready product. Key improvements include integrating the existing system with a mobile robot base, which would allow for autonomous navigation throughout agricultural fields. Adding navigation capabilities to the mobile base is crucial for enabling the robot to maneuver around obstacles and efficiently cover different areas without manual intervention. Additionally, implementing an optimal controller for flow rate control will further refine the precision of pesticide application, ensuring that each crop receives the exact amount needed without overuse or waste, thereby maximizing both efficacy and sustainability. Another significant upgrade involves transitioning from using a conventional laptop for

system control to a more robust and compact single-board computer. This shift will not only reduce the overall size and weight of the control system but also enhance its reliability and power efficiency, making the robot more practical and scalable for commercial use. These enhancements will not only improve the operational capabilities of the robot but also ensure its adaptability and effectiveness in diverse agricultural settings, paving the way for broader commercial adoption and implementation.

1 Introduction and Problem Statement

1.1 Introduction

This report presents a detailed overview of a precision pesticide spraying robot designed to optimize pesticide application through targeted spraying. Utilizing innovative technologies such as a depth camera and sophisticated image recognition models, this robot aims to improve crop health and yield while minimizing the use of chemicals.

The core of this robotic system is its ability to accurately detect and analyze crops using a depth camera. This camera facilitates precise identification and mapping of crop areas, essential for effective pesticide application. To enhance this capability, we developed two distinct image recognition models. The first model is trained to calculate the surface area of the identified crops, which is crucial for determining the optimal spraying distances based on a theoretical coverage formula. The second model focuses on detecting lesions on leaves, allowing for targeted treatment of diseased plants, thereby preventing the wasteful application of chemicals on healthy foliage.

Once the crop and its condition are accurately assessed, the system computes the position coordinates of the target area. These coordinates guide the robotic arm, ensuring it positions itself at an optimal distance from the crop for effective pesticide application. The final stage involves the precise control of the spraying mechanism, where the nozzle and pump are activated to deliver the pesticide only where it is needed.

1.2 Problem Statement

Pesticides are important to crop management because they contribute to increasing crop yields and improving the quality of crops. The conventional methods of pesticide application suffer from inherent inefficiencies, including imprecise targeting leading to drift and excessive chemical usage. These issues not only result in environmental concerns but also contribute to reduced efficacy and increased costs for

farmers. Therefore, there is a pressing need to develop a more precise and efficient solution that addresses these challenges in pesticide application.

1.3 Preliminary Design Goals

In our design, the depth camera is used for crop detection, utilizing image recognition to measure the surface area of captured plants or to assess the area of lesions on leaves to distinguish the health status of the plants. The system then transmits the location of the target crops to the robotic arm, which precisely moves to the specified spray position for accurate spraying. Concurrently, the computer calculates the required spraying distance based on the plant's surface area or the lesion area on the leaves, ensuring complete coverage of the entire plant or affected leaves and the accuracy of the pesticide dosage. This information is transmitted to the spray control system, which is executed by the nozzles to accomplish precise pesticide application.

To pursue excellence, we have set specific objectives for each critical component of the testing process. For the image recognition stage, our focus is on measuring the model's accuracy and recall in correctly identifying healthy and diseased leaves, as well as the precision in measuring the total surface area of the plants. The most important goal is to ensure that the model's mean average precision (mAP) exceeds 0.5, indicating high accuracy.

In the field of robotic arm movement testing, precision is crucial. Our goal is to ensure that the error in the robotic arm's movement to the target spraying position during the spraying process remains within 10%, achieving exceptional precision.

Furthermore, as part of our commitment to sustainable practices, we have set a goal to reduce pesticide waste. Our objective is to achieve a 20% reduction in pesticide waste per unit area, which indicates significant progress in environmental responsibility compared to traditional methods of pesticide application.

2 Background Research and Literature Review

2.1 Detail Background and Market Research

A precise atomization uniform pesticide application control system for a greenhouse was developed, which includes a medicine mixing module, an aerosol delivery module, a detection induction module, and a control module [1].

A greenhouse spraying robot control system was disclosed, which includes a driving device, an anti-collision detection device, an on-track detection sensor, a distance measuring device, a controller, and a spraying device [2].

An automatic spraying robot for a greenhouse was developed, which includes a track [3].

These existing solutions have made significant contributions to the field. However, they may have limitations such as lack of precision, inability to adapt to different types of crops or diseases, or high cost. Therefore, there is still room for improvement and innovation.

The historical context of this project lies in the rapid development of agricultural robotics in recent years, driven by the need for precision agriculture, resource saving, improvement of safety conditions, and shortage of human labor. The societal context is the increasing demand for food production due to the growth of the world population, the need for environmental sustainability, and the challenge of producing more food with less land [4].

Several scientific articles related to this project have been found. For example, one article discusses a smart and novel electric sprayer that can be assembled on a robot. The sprayer has a crop perception system that calculates the leaf density based on a support vector machine (SVM) classifier [5]. Another article provides a comprehensive discussion of the state-of-art of robotic spraying, with a review of weed and disease sensing tasks, and of precision actuation of treatments [6]. These ideas could help improve the precision and efficiency of the spraying robot in your project.

2.2 Customer Needs

Our target customers are greenhouse growers. As engineering students, we found it challenging to interact directly with professionals in the greenhouse-growing field. To bridge this gap, we consulted with SIUE crop science professor Carolyn Butts-Wilsmeyer, Ph.D., whose expertise was instrumental in helping us identify our clients' needs. Through her guidance, it became clear that our project needed to address several critical aspects to meet the specific demands of precision pesticide application. Firstly, the robot must accurately locate spray targets and apply pesticides efficiently to minimize waste and environmental impact. This includes the capability to assess plant health and deliver pesticides only where necessary. Additionally, to reduce downtime, users require the robot to load and dispense a precise amount of pesticide efficiently, promoting continuous operation with minimal interruptions. In terms of durability and reliability, there is a strong need for products that are robust, long-lasting, and capable of withstanding the rigors of field use. The robot also needs to operate effectively within certain environmental conditions, as users have stressed the importance of components that can adapt seamlessly to fluctuations in temperature and humidity. Moreover, addressing the custom needs of ease of use and full automation, the design ensures that the robot can be operated with minimal manual intervention, making it accessible and user-friendly for greenhouse growers. By integrating these needs, both explicit and latent, our development process is guided by the expectations of greenhouse plant growers, ensuring that our automated systems integrate seamlessly with their operational requirements and preferences.

3 Design

3.1 Preliminary Design Ideas

The initial concept involves employing a two-degree-of-freedom robotic arm with sensors for plant detection and four nozzles for spraying. While this design is cost-effective with efficient manufacturing, it suffers from pesticide wastage and lacks precision in spraying, unable to assess plant health status [*Figure 1J*].

To address these issues, we upgraded to a four-degree-of-freedom robotic arm, incorporated a camera for plant detection, and maintained four nozzles for spraying. This revised design enhances precision by assessing plant health but still encounters pesticide wastage [*Figure 2*].

In our continued efforts to address the issue, we upgraded to a more sophisticated six-degree-of-freedom robotic arm equipped with a plant-detecting camera and reduced the number of nozzles to one, significantly minimizing pesticide waste. Although this design offers flexibility and precise plant health detection, it proved expensive and challenging to manufacture [*Figure 3*].

As a result, we reverted to a four-degree-of-freedom robotic arm configuration, equipped with a camera and a single nozzle at the end effector. This streamlined design successfully identifies plant health, reduces pesticide waste, and maintains accuracy, albeit with slightly less flexibility than the six-degree-of-freedom robotic arm [*Figure 4*].

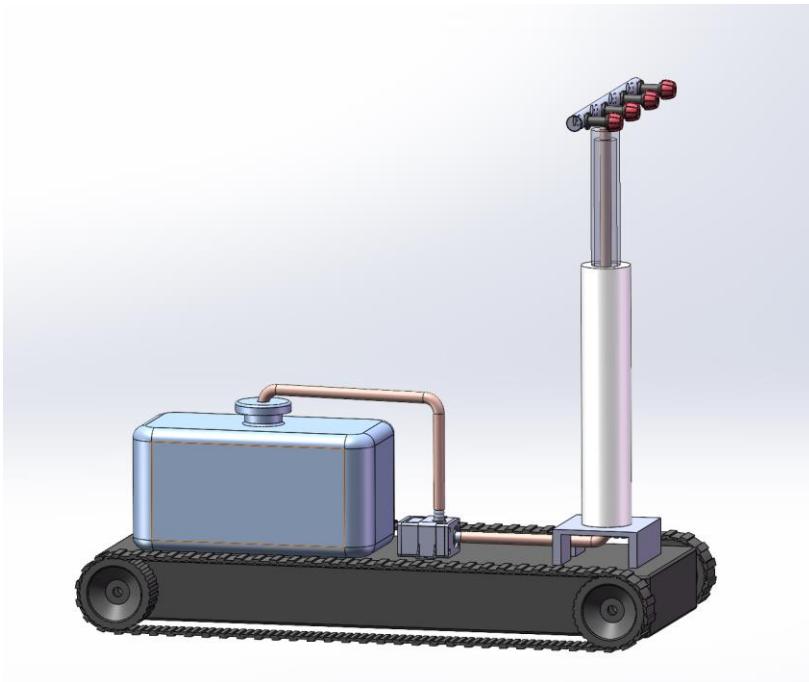


Figure 1 Option 1

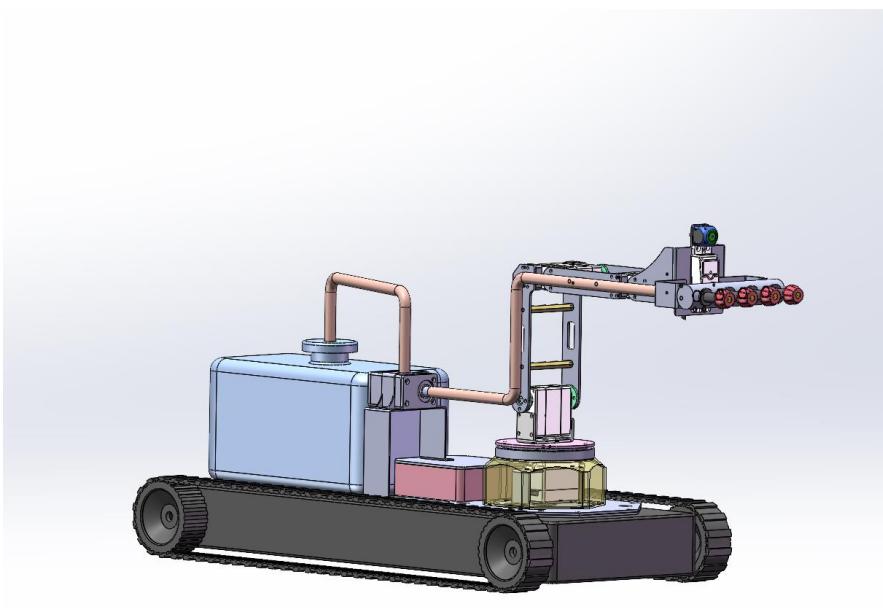


Figure 2 Option 2

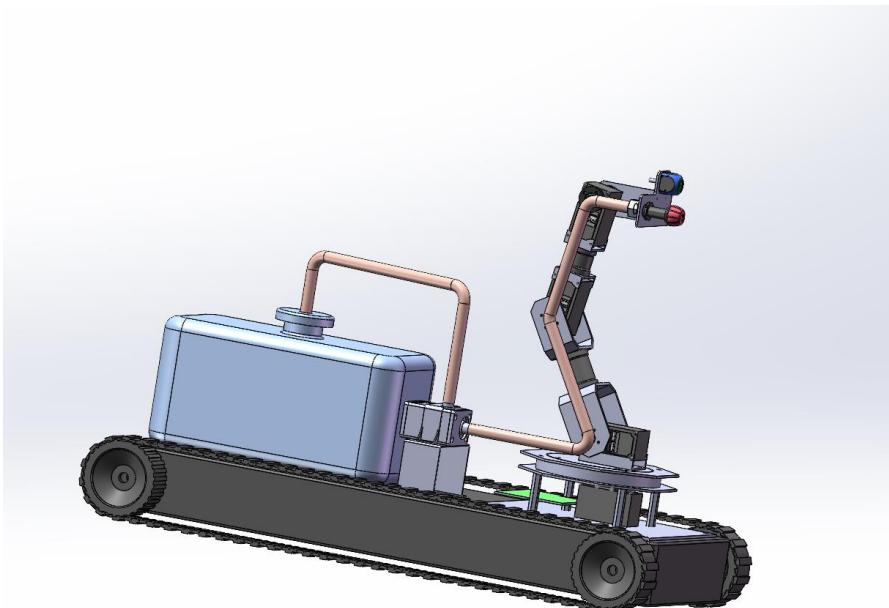


Figure 3 Option 3

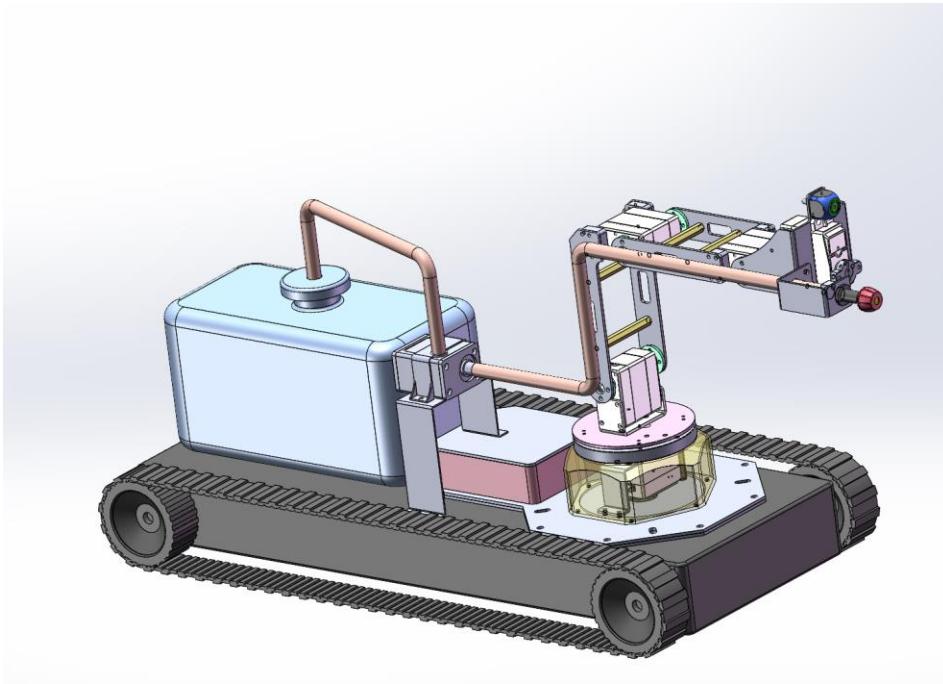


Figure 4 Option 4

3.2 Design trade study

For the trade study, we can see that the option 4 is the optimal choice [Table 1].

Table 1 Trade Study

		Concept							
Criteria	Weight	Opt 1		Opt 2		Opt 3		Opt 4	
		Rating (1-5)	Score						
Requirements									
Ease of use	10%	5	0.5	3	0.3	3	0.3	4	0.4
Precision	15%	1	0.15	3	0.45	3	0.45	3	0.45
Flexibility	15%	1	0.15	3	0.45	4	0.6	4	0.6
Pesticide Saving	20%	1	0.2	3	0.6	4	0.8	4	0.8
Suitable for complex terrain	20%	4	0.8	4	0.8	4	0.8	4	0.8
Working Efficiency	20%	3	0.6	3	0.6	2	0.4	2	0.4
Total	100%		2.4		3.2		3.35		3.45
Rank		5		3		2		1	
Continue?		No		No		No		Yes	

3.3 Target/Final Specifications

Talk About It

Table 2 Target/Final/Alpha/Beta Specifications

Metric	Target specification(design)	Final specification(analyses)		Alpha Test	Beta Test
Overall Weight	15-20kg (overall) 4-5kg (without mobile robot base)	16kg (overall) 4kg (without mobile robot base)		16kg (overall) 4kg (without mobile robot base)	17kg (overall) 5kg (without mobile robot base)
Spray Range (theoretical coverage)	40cm	Spray Angle	25 degrees	~23 degree	~23 degree
		Spray Distance	100cm	150cm (maximum)	150cm (maximum)
		Water Pipe Diameter	6.5mm	4.5mm	4.5mm
		Nozzle Hole Diameter	0.8mm-1mm	0.8mm	0.8mm
Robot Arm DOF	4	4		4	4
Robot Arm Workspace	200mm-300mm (reach) 400mm-800mm (span)	200mm (reach) 400mm (span)		300mm(reach) 600mm(span)	300mm(reach) 600mm(span)
Robot Arm Working Payload	50g-100g	50g		50g (maximum)	50g (maximum)
Robotic Arm Joint Torque	1.5N.m	0.4N.m		1.5N.m	1.5N.m
Tank Capacity	3.5L	2.2L		5.6L	5.6L
Camera Resolution	1920× 1080 pixels	640×480 pixels		1280×720 pixels	1280×720 pixels
Control Method	Automatic	Semi-automatic		Semi-automatic	Semi-automatic
Material	Aluminum Alloy and plastic and latex	Aluminum Alloy and plastic and latex		Aluminum Alloy and plastic and latex	Aluminum Alloy and plastic and latex and wood
Working Time	1-2h(continuous)	2h(continuous)		4h (continuous)	4h (continuous)
Battery	12V 6Ah	n/a		12V 24Ah	12V 24Ah
Pesticide Spray Amount	50-75 gallon per acre	Pressure of pump	0.1-0.45Mpa	<0.5Mpa	<0.5Mpa
		Flow rate of nozzle	0.2-0.3gpm	0.1-0.6L/min	0.5-0.7L/min
Ground Speed	0-10Km/hr	0-8.5Km/hr		n/a	n/a
Cost	800	700		600	700

3.4 Product Architecture

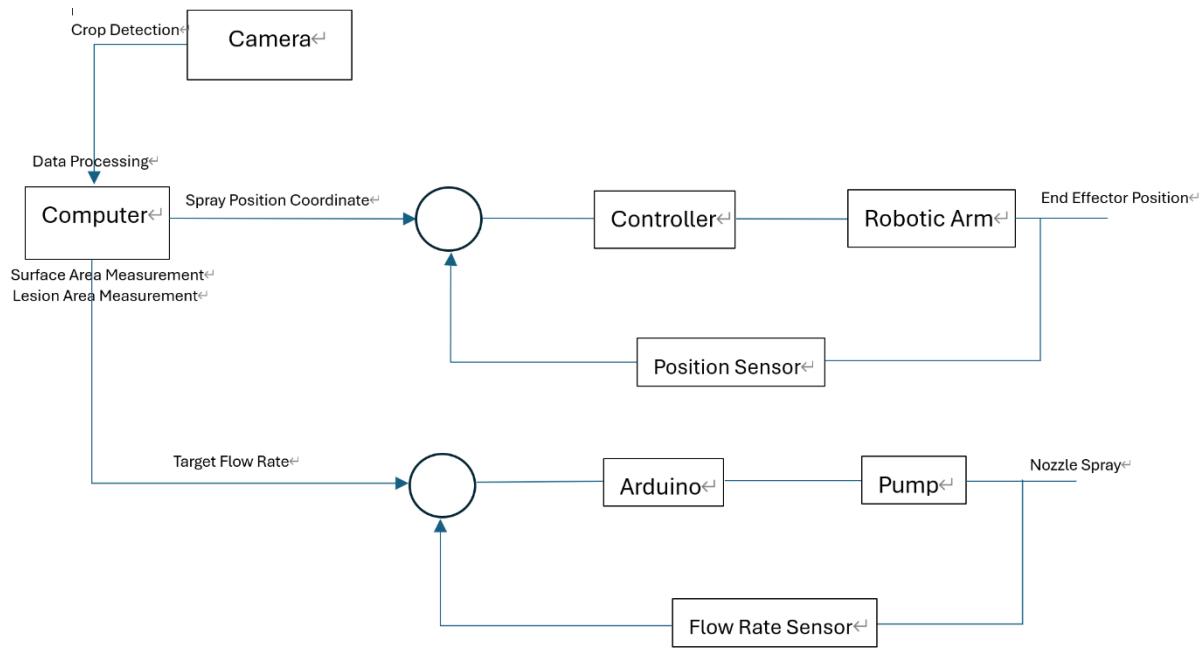


Figure 5 Block Diagram

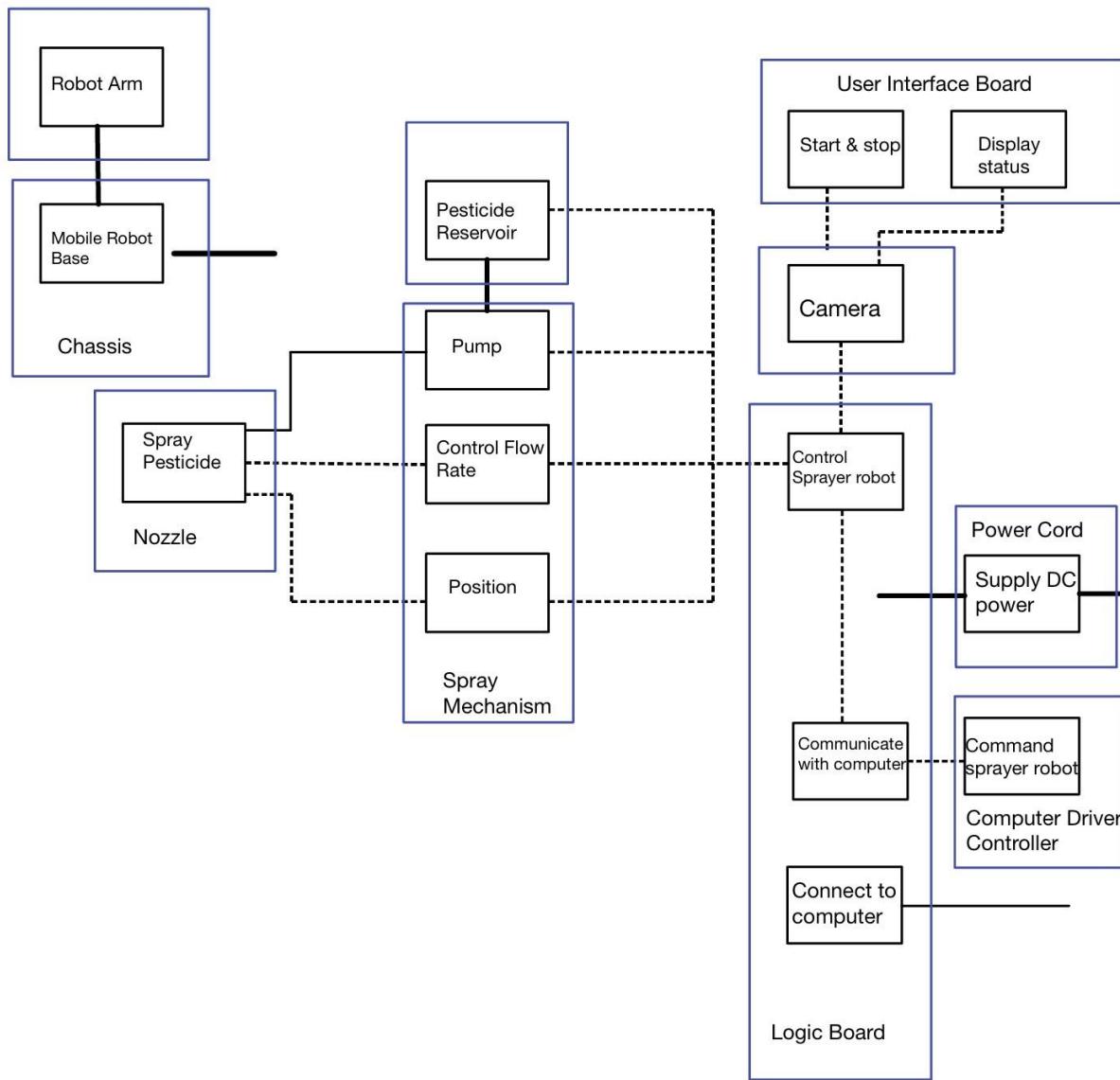


Figure 6 Product Architecture

The crop detection module uses a camera to capture crop images and is connected to the image recognition module (on the logic board). The image recognition module (located on the logic board) analyzes the captured images to assess crop health or measure the surface area of each plant. Receives input from the crop detection module and transmits analytical data to the robotic arm control module. The robotic arm control module receives the position information of the target crop and the measured surface area data to calculate the appropriate spraying distance and control the movement of the robotic arm. Connect the image recognition module to obtain target crop position information and connect the pesticide application

control module for synchronization. Pesticide application control module controls the flow rate of pesticides. Receive information from the robotic arm control module for spray flow rate control and control of nozzle start and pause. Nozzle performs spraying tasks according to instructions from the pesticide application control module.

The crop detection module provides input to the image recognition module. The image recognition module sends data to the robotic arm control module and pesticide application control module. The robot arm control module controls the movement of the robot arm and moves it to a specified position. The spray control module controls the spray flow rate and controls the start of the spraying task.

3.5 Discussion of Constraints, Risks and Back-up Plans*

The limitation of our product is its incompatibility with extensive farmland areas characterized by a diverse range of crops. The project poses potential waterproofing risks, as the practical application may lead to issues such as nozzle leakage and water pipe ruptures, resulting in unintended liquid leakage. If there are no good waterproof measures, liquid may enter the motor or penetrate into the circuit, resulting in a safety hazard on the board. To mitigate the liquid-leakage risks, we plan to encase each joint of the robot arm with waterproof material. Additionally, we intend to centralize the circuitry in a secure and concealed location, strategically positioned away from the path of liquid flow. What is more, if the camera and nozzle installed on the robot arm are too heavy, the inertia of the robotic arm can lead to a loss of accuracy and droplet drift. We will consider multiple types of cameras and different sizes and shapes of nozzles and try to use lighter materials (latex and plastic) for the nozzle and water pipes. For high-risk aspects, we have considered preparing different materials and assembling them for actual testing. If none of them are suitable for the project, we will consider using 3D printing to design the parts we want.

4 Analysis

4.1 Analysis Plans

For the analysis, we have selected robot arm analysis, camera analysis, and nozzle and pump analysis. We consider these to be the most relevant and critical aspects of our project, requiring the utmost time and effort for a thorough examination. The primary analysis is focused on the camera, as the product is designed to accurately identify diseased leaves and determine their location for targeted spraying. The functioning of the robotic arm and nozzle is contingent upon the proper operation of the camera. Therefore, we prioritize camera analytics as our top concern.

The subsequent analysis is dedicated to the robotic arm to ensure it meets our specific requirements. We must verify that the chosen robotic arm can reach the target position and that the joint motor torque is sufficient to support the loaded robotic arm in reaching the designated position within two seconds. This analysis holds significant importance, given that the robotic arm is a crucial component of the product, responsible for moving both the nozzle and camera.

Following the robotic arm analysis, we move on to the nozzle and water pump examinations. In this phase, our goal is to confirm that the pressure and flow rate within the system align with the spray angle and distance requirements of the nozzle. Additionally, we aim to establish the product specifications for the water pump. Once again, we emphasize the importance of this analysis, as the proper functioning of the nozzle is contingent upon the pump providing sufficient pressure. Each analysis chosen is considered critical to the success of our product. Through a meticulous examination of every aspect, we aim to fine-tune our design to meet our specified requirements.

The forthcoming analyses are integral components of our planned assessments. First, the battery analysis aims to calculate the precise battery capacity required for the continuous operation of our product, providing insights into its overall runtime. Additionally, the nozzle analysis seeks to resolve the anti-drip issue, enhancing the precision and efficiency of the spraying process. Lastly, the integrated control analysis

is paramount for seamlessly integrating the control systems of the robotic arm, camera, nozzle, and pump. This integration is crucial for ensuring synchronized and efficient operation across all components of our automated spraying system. Together, these analyses form essential steps in refining and optimizing our product design, addressing potential challenges, and improving overall functionality and reliability.

4.2 Robotic Arm Analysis

The physical problem is ensuring that the target plants can be sprayed. Tomato plants in the greenhouse are typically spaced 16 inches (0.4 meters) apart, with plant heights ranging from 60 cm to 1.5 meters (varieties of tomatoes will have varying plant heights). This is all dependent on the needs of our customers. Next, the assumption is made that the robotic arm can be positioned in the middle of the space between plants and that its operational range can encompass all the tomato plant's leaves.

The end effector of the robotic arm doesn't need to reach the target plant all the time because there will be a nozzle at the end of the robotic arm, and the liquid sprayed from the nozzle will also have a certain distance, but this distance will not be very far, about 5–10 cm or so, because if the distance is very far, it cannot spray accurately. So, to simplify this problem, we just need a part of the working range of the end effector of the robotic arm to be able to reach the target plant, and the rest of the plant needs a nozzle to cooperate with the robotic arm for spraying. Finally, the assumption is made that the robotic arm's operating space should reach 20 cm by 40 cm.

Through the above product design, it has been confirmed that the four-degree-of-freedom robotic arm is the main body of the designed product, so frames are first established based on a simple diagram [*Figure 7*].

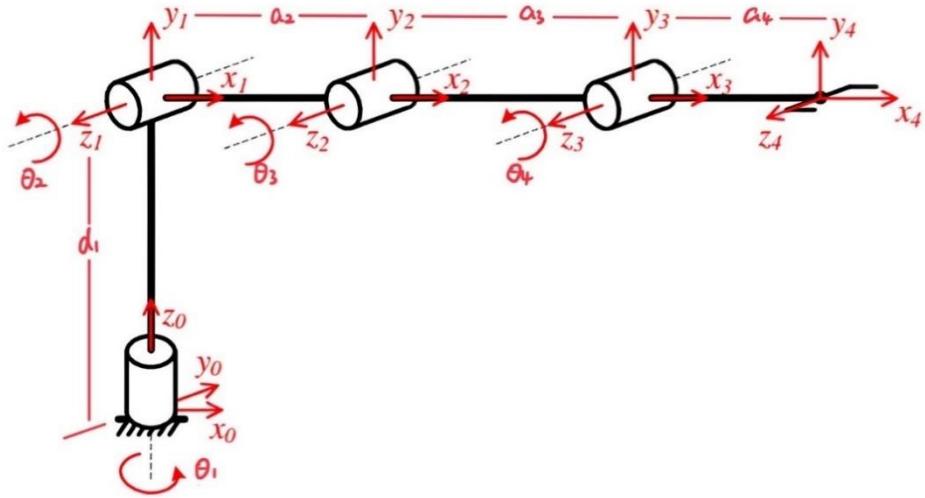


Figure 7 Denavit Hartenberg Convention

Then the Denavit–Hartenberg (DH) parameters table of the four-degree-of-freedom manipulator is obtained [Table 4.2.1].

Table 0.1

Link	θ	d	a	α
1	θ_1^*	d_1	0	$\frac{\pi}{2}$
2	θ_2^*	0	a_2	0
3	θ_3^*	0	a_3	0
4	θ_4^*	0	a_4	0

Substituting the DH parameters into homogeneous transformation matrices, four matrices are obtained.

$${}^{i-1}T = \begin{pmatrix} \cos(\theta_i) & -\cos(\alpha_i) \sin(\theta_i) & \sin(\alpha_i) \sin(\theta_i) & a_i \cos(\theta_i) \\ \sin(\theta_i) & \cos(\alpha_i) \cos(\theta_i) & -\sin(\alpha_i) \cos(\theta_i) & a_i \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.2.1)$$

$${}_1^0T = \begin{pmatrix} \cos(\theta_1) & 0 & \sin(\theta_1) & 0 \\ \sin(\theta_1) & 0 & -\cos(\theta_1) & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.2.2)$$

$${}_2^1T = \begin{pmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & a_2 \cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & a_2 \sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.2.3)$$

$${}_3^2T = \begin{pmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & a_3 \cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & a_3 \sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.2.4)$$

$${}_4^3T = \begin{pmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & a_4 \cos(\theta_4) \\ \sin(\theta_4) & \cos(\theta_4) & 0 & a_4 \sin(\theta_4) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.2.5)$$

Then multiply these four adjacent transformation matrices to obtain the overall transformation matrix ${}_4^0T$.

$${}_4^0T = {}_1^0T {}_2^1T {}_3^2T {}_4^3T \quad (4.2.6)$$

$${}_4^0T = \begin{pmatrix} R(\theta) & p(\theta) \\ 0 & 1 \end{pmatrix} \quad (4.2.7)$$

$$R(\theta) = \begin{bmatrix} \cos(\theta_2 + \theta_3 + \theta_4) \cos(\theta_1) & -\sin(\theta_1) & -\sin(\theta_2 + \theta_3 + \theta_4) \cos(\theta_1) \\ \cos(\theta_2 + \theta_3 + \theta_4) \sin(\theta_1) & \cos(\theta_1) & -\sin(\theta_2 + \theta_3 + \theta_4) \sin(\theta_1) \\ \sin(\theta_2 + \theta_3 + \theta_4) & 0 & \cos(\theta_2 + \theta_3 + \theta_4) \end{bmatrix} \quad (4.2.8)$$

$$p(\theta) = \begin{bmatrix} \cos(\theta_1) [a_3 \cos(\theta_2 + \theta_3) + a_2 \cos(\theta_2) + a_4 \cos(\theta_2 + \theta_3 + \theta_4)] \\ \sin(\theta_1) [a_3 \cos(\theta_2 + \theta_3) + a_2 \cos(\theta_2) + a_4 \cos(\theta_2 + \theta_3 + \theta_4)] \\ d_1 + a_3 \sin(\theta_2 + \theta_3) + a_2 \sin(\theta_2) + a_4 \sin(\theta_2 + \theta_3 + \theta_4) \end{bmatrix} \quad (4.2.9)$$

Upon the determination of, ${}_4^0T$ we can find the global coordinates of the end effector. The tip point of the arm is at the origin of frame 4, i.e., it is at $[0 \ 0 \ 0 \ 1]^T$. So, its position in the global frame becomes:

$${}_p^0r = {}_4^0T {}_p^4r = {}_4^0T \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} r_{14} \\ r_{24} \\ r_{34} \\ 1 \end{bmatrix} = \begin{bmatrix} dx \\ dy \\ dz \\ 1 \end{bmatrix} \quad (4.2.10)$$

Obviously, this is the last column of the transformation matrix ${}_4^0T$. After simplifications using trigonometric formulas, the previous equation becomes:

$$dx = \cos(\theta_1) a_3 \cos(\theta_2 + \theta_3) + a_2 \cos(\theta_2) + a_4 \cos(\theta_2 + \theta_3 + \theta_4) \quad (4.2.11)$$

$$dy = \sin(\theta_1) a_3 \cos(\theta_2 + \theta_3) + a_2 \cos(\theta_2) + a_4 \cos(\theta_2 + \theta_3 + \theta_4) \quad (4.2.12)$$

$$dz = d_1 + a_3 \sin(\theta_2 + \theta_3) + a_2 \sin(\theta_2) + a_4 \sin(\theta_2 + \theta_3 + \theta_4) \quad (4.2.13)$$

In addition, dx , dy and dz are the global end effector coordinates. Moreover, the end effector orientation is:

$$\varphi = \theta_2 + \theta_3 + \theta_4 \quad (4.2.14)$$

Four nonlinear equations with four unknowns are left with. For a given end effector location $[dx, dy, dz]$ and orientation φ , we need to know the joint variables $\theta_1, \theta_2, \theta_3, \theta_4$ in order to solve these equations algebraically, a process known as inverse kinematics. By squaring, dividing, adding, and applying a few trigonometric formulas, we arrive at the following equation:

$$\theta_1 = \tan^{-1} \left(\frac{dy}{dx} \right) \quad (4.2.15)$$

$$\theta_2 = \tan^{-1} \left(c, \pm \sqrt{r^2 - c} \right) - \tan^{-1}(a, b) \quad (4.2.16)$$

$$\theta_3 = \cos^{-1} \left(\frac{A^2 + B^2 + C^2 - a_2^2 - a_3^2}{2a_2 a_3} \right) \quad (4.2.17)$$

Where $a = a_3 \sin(\theta_3)$, $b = a_2 + a_3 \cos(\theta_3)$, $c = dz - d_1 - a_4 \sin(\varphi)$, and $r = \sqrt{a^2 + b^2}$

In addition, $A = (dx - a_4 \cos(\theta_1) \cos(\varphi))$ $B = (dy - a_4 \sin(\theta_1) \cos(\varphi))$ $C = (dz - d_1 - a_4 \sin(\varphi))$

Having determined $\theta_1, \theta_2, \theta_3$, we can then find θ_4 from the EE orientation of φ as follows:

$$\theta_4 = \varphi - \theta_2 - \theta_3 \quad (4.2.18)$$

ADAMS is selected as the simulation software because it establishes system dynamics equations and performs statics and motion on the virtual mechanical system using the Lagrange equation method in the multi-rigid body system dynamics theory. Force curves can be used for reaction, displacement, acceleration, and velocity in mechanical and dynamic analysis. This verification process and its findings, in my opinion, are ideal for our technical analysis simulation.

Set the simulation parameter coordinates to move the end effector of the manipulator to the target position [Figure 4.2. 2].

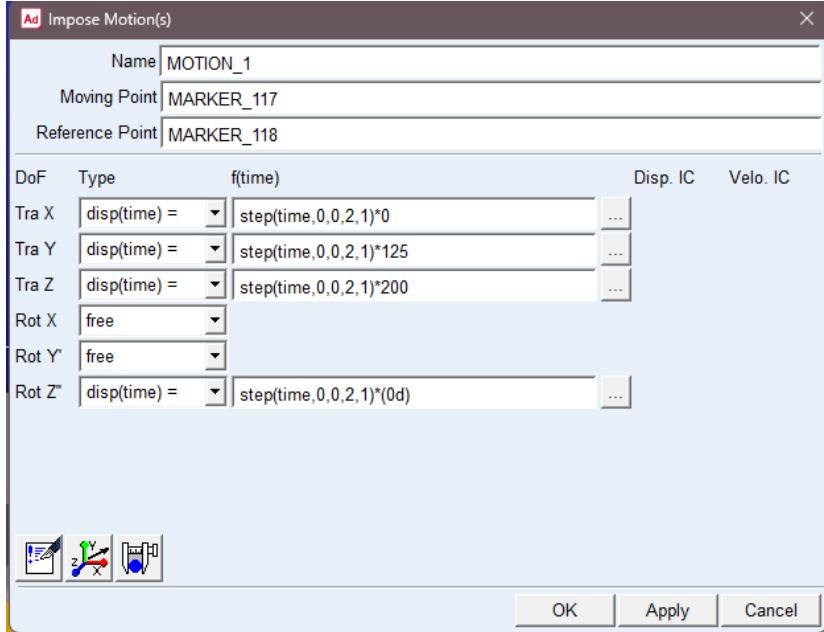


Figure 0.8

Through simulation, the end effector of the robotic arm can reach the specified position, which means that the assumed workspace can be satisfied [Figure 4.2.3].

In the next semester, one of the requirements to consider when choosing a robotic arm is that the working space must not be less than reach 20cm and span 40cm.

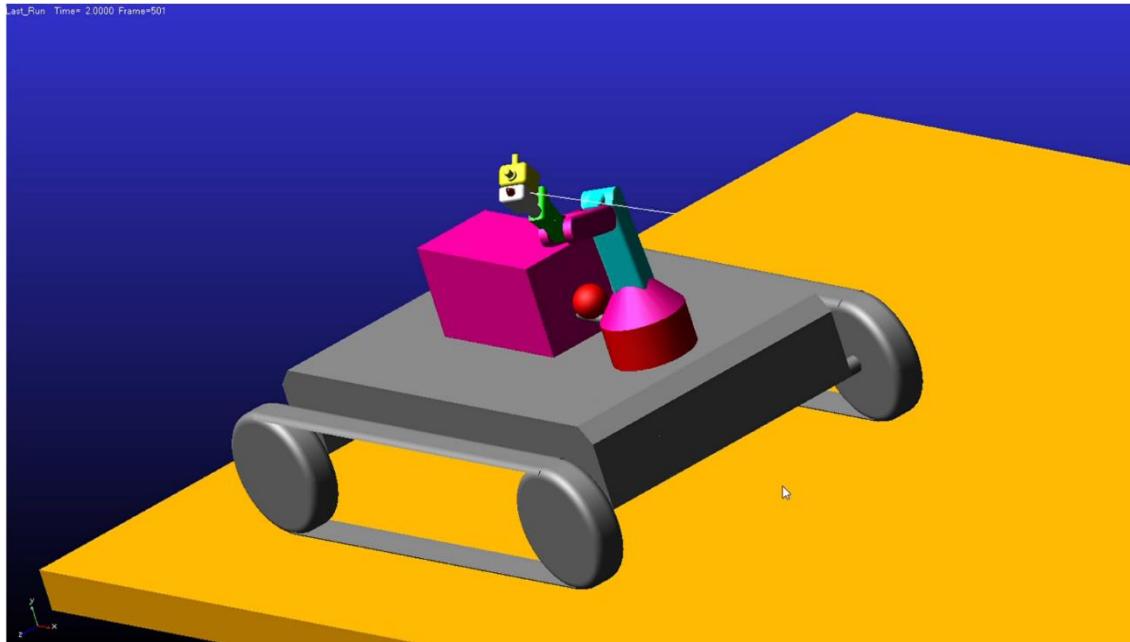


Figure 0.9

When selecting a robotic arm, you should not only consider the working space, but also consider the torque of the joint motor of the robotic arm. If the torque of the motor is insufficient, the robotic arm will not work properly. Because this product is designed to have a camera recognition function and a nozzle spray function, a camera and a nozzle will be installed at the end of the robotic arm, so the gravity of the camera and nozzle must be considered. As we all know, there is a reaction force when the nozzle sprays. Although it is small, it still needs to be considered.

Of course, there will be other influencing factors, such as friction and so on. So, in this analysis, assuming that the end of the robotic arm only has a gravity of 50g from the camera and nozzle, and a reaction force of 0.2N from a nozzle, the robotic arm can reach the designated position within two seconds under the action of these forces. This general formulation of the governing dynamical equation of motion will vary depending on the Lagrangian principle that seeks to derive it.

$$\tau = M\ddot{\theta} + V + G, \tau = Q - J^T F_e + f_r \quad (4.2.19)$$

The inertia forces are the first term, the Coriolis and centrifugal forces are the second, and the gravitational effects forces are represented by the third term. Where τ , $\ddot{\theta}$, J^T , F_e , f_r stand for joint torque,

joint acceleration, the force acting on the joint, the Jacobian transpose, and friction force, in that order. The first term of equation (4.2.19) that represents the inertia matrix is shown in (4.2.20).

$$M(\theta) = J_{v1}^T m_1 J_{v1} J_{v2}^T m_2 J_{v2} + J_{v3}^T m_3 J_{v3} J_{v4}^T m_4 J_{v4} + J_{w1}^T I_1 J_{w1} + J_{w2}^T I_2 J_{w2} + J_{w3}^T I_3 J_{w3} + J_{w4}^T I_4 J_{w4} \quad (4.2.20)$$

We might obtain the 4×4 matrix for the first term of the general version of the dynamic equation by carrying out the computations illustrated in (4.2.20).

$$M_{ij}(\theta) = \begin{pmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{21} & M_{22} & M_{23} & M_{24} \\ M_{31} & M_{32} & M_{33} & M_{34} \\ M_{41} & M_{42} & M_{43} & M_{44} \end{pmatrix} \quad (4.1.21)$$

Furthermore, Centrifugal forces will be computed using the Christoffel Symbols of the First Kind in the second term of (4.2.19) that represents Coriolis. This will simplify and improve the accuracy of the calculations when we use them to compute torque.

$$c_{ijk} = \frac{1}{2} \left\{ \frac{\partial M_{kj}}{\partial q_i} + \frac{\partial M_{ki}}{\partial q_j} + \frac{\partial M_{ij}}{\partial q_k} \right\} \quad (4.2.22)$$

In a reference frame that is rotating with respect to an inertial frame, this type of force is produced. Furthermore, in a non-inertial frame of reference, objects moving in a circular motion are affected by a "fictitious" force called the centrifugal force. Calculations for c'_{ijk} s are entirely dependent on i, j, k factors. Consequently, the N elements of it were computed using the equation $c_{ijk}_N = 4^3 = 64$ elements, as we have three factors pertaining to robot arm joints: i, j, k . There are four cases of each factor overall. Using (4.2.22), we were able to produce 64 values for c_{ijk} , which we then arranged as indicated in *Table 4.2.2*. We are able to create the final 4×4 Coriolis, Centrifugal Forces Matrix, that will be used in torque computation.

Table 0.2

Column1	
Row1	C111+C121+C131+C141
Row2	C112+C122+C132+C142
Row3	C113+C123+C133+C143

Row4	C114+C124+C134+C144
Column2	
Row1	C211+C221+C231+C241
Row2	C212+C222+C232+C242
Row3	C213+C223+C233+C243
Row4	C214+C224+C234+C244
Column3	
Row1	C311+C321+C331+C341
Row2	C312+C322+C332+C342
Row3	C313+C323+C333+C343
Row4	C314+C324+C334+C344
Column4	
Row1	C411+C421+C431+C441
Row2	C412+C422+C432+C442
Row3	C413+C423+C433+C443
Row4	C414+C424+C434+C444

$$V_{ij}(\theta, \dot{\theta}) = \begin{pmatrix} V_{11} & V_{12} & V_{13} & V_{14} \\ V_{21} & V_{22} & V_{23} & V_{24} \\ V_{31} & V_{32} & V_{33} & V_{34} \\ V_{41} & V_{42} & V_{43} & V_{44} \end{pmatrix} \quad (4.2.23)$$

Moreover, we rewrite the general mechanical equation in a way that makes it appropriate for computing the final torque, refer to (4). This formula serves as the formula for calculating torque. It involves adding up two 4×4 matrices: the inertia matrix (represented by *Mmatrix* and *Vmatrix*) and the matrix of centrifugal forces (represented by Coriolis and matrix) for Coriolis. The robotic arm links length, masses in terms of $\theta, \dot{\theta}$, and $\ddot{\theta}$, and the joint torques regarding these variances regarding time-variant will be the final result of torque.

$$\sum_j m_{kj}(\theta) \ddot{\theta}_j + \sum_{i,j} c_{ijk}(\theta) \dot{\theta}_i \dot{\theta}_j + g_k(\theta) = \tau_k \quad (4.2.24)$$

$k = 1, 2, 3, 4$. On the right-hand side, is the final torque of the joints.

ADAMS is still the software used for simulation. Set the parameters to move the end of the robotic arm to any position within the working range, and then obtain a graph of the torque changes of the four joint motors of the robotic arm over time and obtain the maximum torque among them.

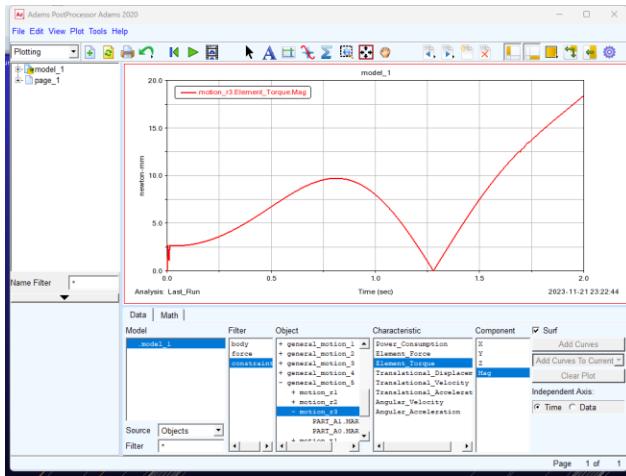


Figure 0.10

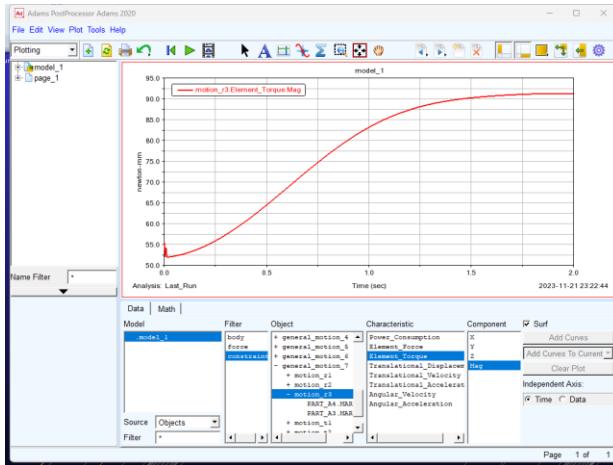


Figure 0.11

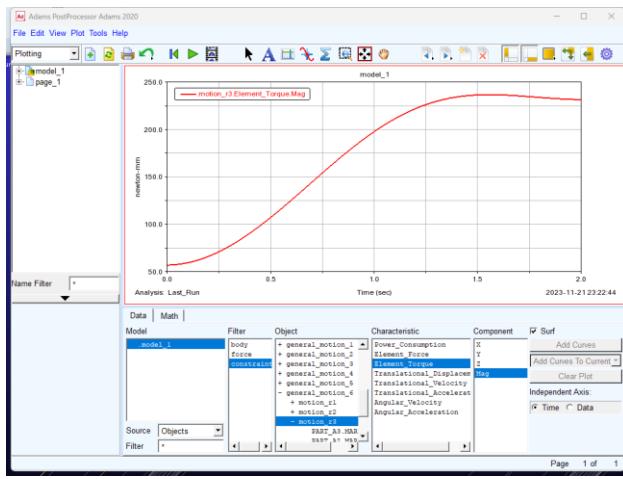


Figure 0.12

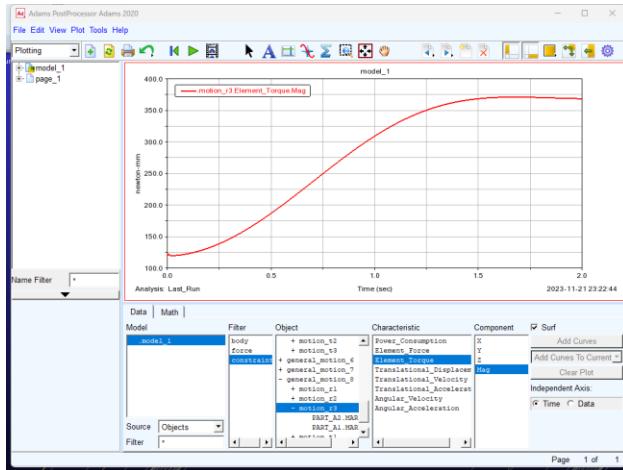


Figure 0.7

Set different parameters to conduct simulation and select several representative sets of data to make

Table 4.2.3.

Table 0.3

θ_1	θ_2	θ_3	θ_4	Maximum torque of joint 1 (N.mm)	Maximum torque of joint 2 (N.mm)	Maximum torque of joint 3 (N.mm)	Maximum torque of joint 4 (N.mm)	Maximum of all torque (N.mm)
123°	171°	16°	61°	8.8	429	191	193	429
18°	-48°	-120°	-81°	0.6	211	183	93	211
-77°	15°	111°	81°	3.2	414	191	186	414
134°	140°	-25°	59°	7.8	425	189	191	425
75°	46°	-13°	157°	3.1	419	189	189	419
-176°	-156°	159°	20°	17.1	363	177	163	363
164°	121°	75°	-161°	15.9	413	190	183	413
178°	12°	48°	-49°	12	374	175	167	374
-66°	107°	-72°	-28°	2.7	416	188	186	416
-136°	107°	-29°	-44°	5.6	416	188	186	416
39°	-177°	-125°	-147°	5.3	376	270	167	376
11°	112°	-132°	103°	1.9	416	122	102	416
-136°	-144°	-20°	118°	20.3	395	187	178	395
150°	150°	150°	150°	31	465	239	195	465

From the above *Table 4.2.3*, it can be clearly seen that the torque of the second joint is the largest of the four joint torques, and the maximum torque in these data is 465N.mm, so it can be concluded that the robotic arm in the case of loads and recoil, the joint motor torque is only about 500N.mm will be enough, so the next semester of the purchase of the robotic arm, the torque of the joint motors of the robotic arm must be greater than 500N.mm.

4.3 Camera and Image Recognition Analysis

It is necessary to identify the health status of leaves to confirm whether there are leaf lesions and then decide whether to spray pesticides. After comparing multiple potentially feasible image recognition models, like YOLO, SSD, Faster R-CNN, and so on, YOLOv5 is chosen to be the leaf lesion recognition model. Its ease of use, good algorithm, and very fast recognition speed are its main advantages.

4.3.1 Model Performance Evaluation

Figure 4.3.1 shows the overall performance of the model currently used. The model is trained by using the standard yolov5s.pt model with 400 epochs. The dataset has 2330 images as training samples. For more details [7].

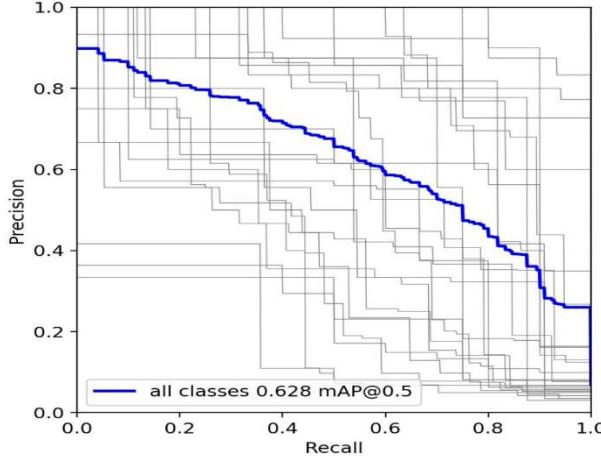


Figure 0.14

mAP is used to evaluate performance of the model. Because this is a model for detecting multiple types of crop leaf lesions, it is necessary to consider each type of leaf lesions' detection precision rate.

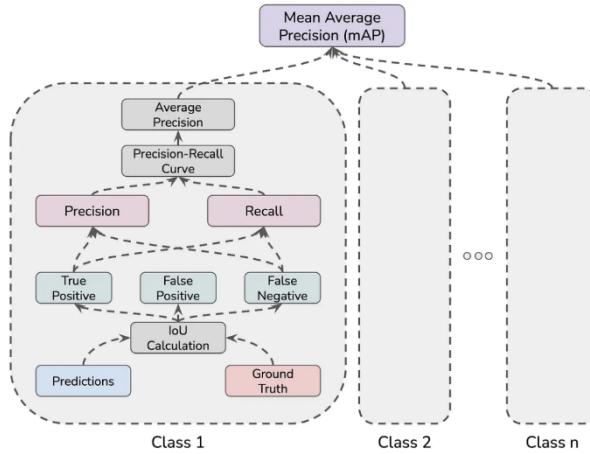


Figure 0.15

Figure 4.3.2 [8] shows the process of how mAP is calculated, here are the formulas needed:

$$\text{Precision} = \frac{\text{True Positive}(TP)}{\text{True Positive}(TP) + \text{False Positive}(FP)} \quad (4.3.1)$$

$$\text{Recall} = \frac{\text{True Positive}(TP)}{\text{True Positive}(TP) + \text{False Negative}(FN)} \quad (4.3.2)$$

$$AP = \int_{r=0}^1 p(r)dr \quad (4.3.3)$$

$$mAP = \frac{1}{N} \sum_{i=0}^N AP_i \quad (4.3.4)$$

The current mAP of the model used is 0.628, the standard the YOLOv5s model mAP is 0.568 [9]. So, performance of model the is acceptable and will continue to be improved. Potential ways to improve performance are to increase the number of dataset images, improve sample images qualities, or try different training epochs.

4.3.2 Distance estimation

After identification of leaf lesions. The robotic arm needs to obtain the specific location of the target from the camera to move. Common cameras cannot provide depth information for the z-axis. But depth cameras capture not only color and intensity like the common cameras but also provide depth information. This additional depth data provides a third dimension, allowing the camera to perceive the distance of objects from the camera.

Intel D435i Depth Camera is used in this design. The D435i includes factory-calibrated sensors, which simplifies the integration process and ensures accurate depth. It also has a depth range of approximately 0.2 to 10 meters, making it suitable for various depth-sensitive applications. For this design, assume measuring the distance of object is within 1 meter and the error shall not exceed 5%.

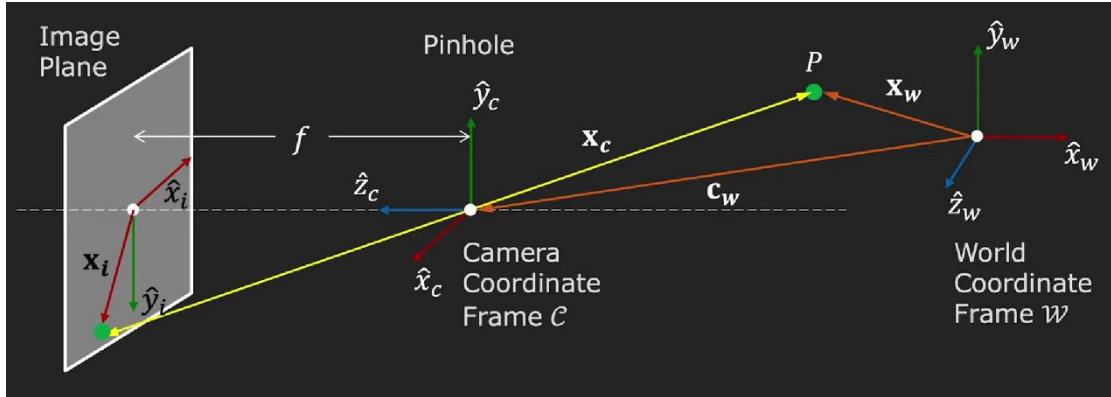


Figure 0.16 Position calculation by using principle of triangulation.

From the *Figure 4.3.3* lists vectors about image coordinates and camera coordinates:

$$\vec{x}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}, \vec{x}_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \quad (4.3.5)$$

$$\frac{x_i}{f} = \frac{x_c}{z_c}, \frac{y_i}{f} = \frac{y_c}{z_c} \quad (4.3.6)$$

$$x_i = f * \frac{x_c}{z_c}, y_i = f * \frac{y_c}{z_c} \quad (4.3.7)$$

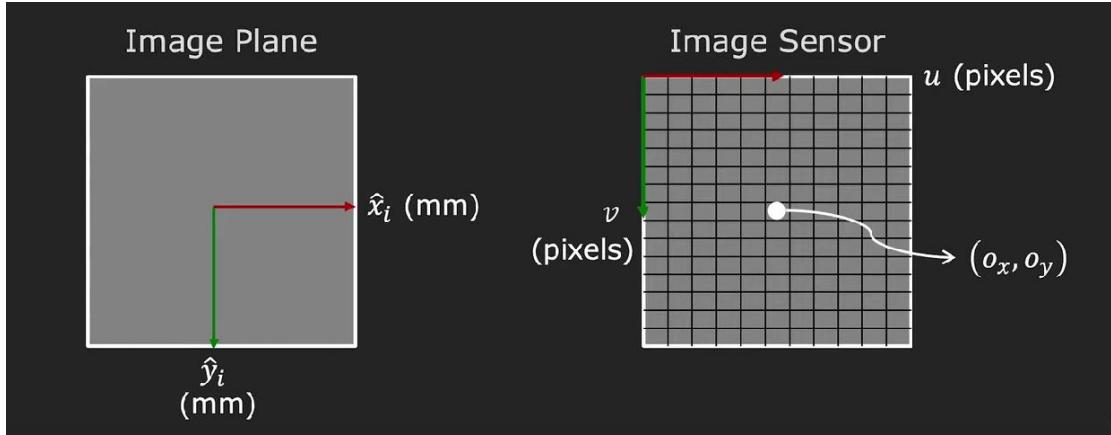


Figure 0.17 Pixels shape factors

Since pixels can be square or rectangle, assume p_x and p_y are pixel densities in x and y direction.

$$u = p_x * x_i + o_x = p_x * f * \frac{x_c}{z_c} + o_x \quad (4.3.8)$$

$$v = p_y * y_i + o_y = p_y * f * \frac{y_c}{z_c} + o_y \quad (4.3.9)$$

Let $f_x = p_x * f$, $f_y = p_y * f$ (f_x, f_y are focal length in pixels)

$$u = f_x * \frac{x_c}{z_c} + o_x \quad (4.3.10)$$

$$v = f_y * \frac{y_c}{z_c} + o_y \quad (4.3.11)$$

Here, o_x, o_y, f_x, f_y are intrinsic parameters of the camera, representing the camera's internal geometry.

Assume the camera has been calibrated, o_x, o_y, f_x, f_y, b is known.

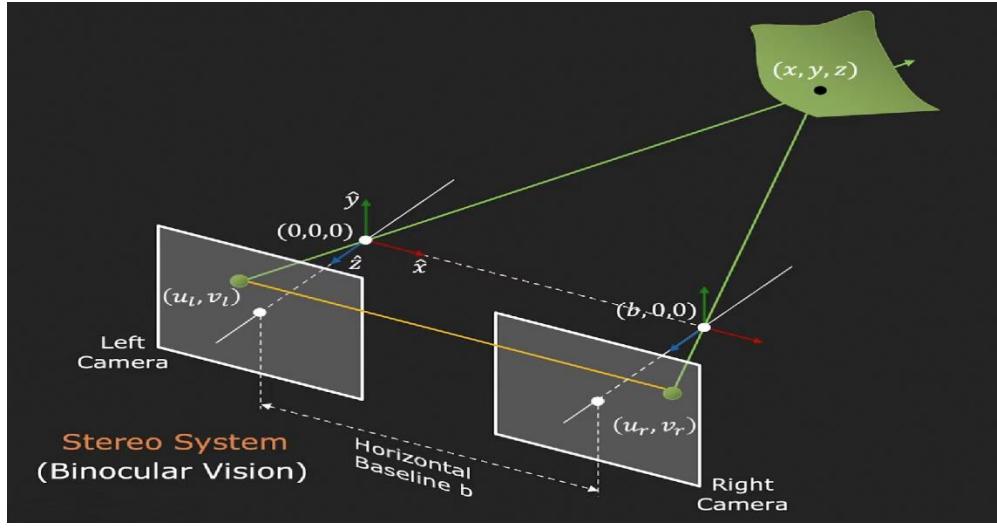


Figure 0.18

$$u_L = f_x * \frac{x}{z} + o_x, u_R = f_x * \frac{x - b}{z} + o_x \quad (4.3.12)$$

$$v_L = f_y * \frac{y}{z} + o_y, v_R = f_y * \frac{y}{z} + o_y \quad (4.3.13)$$

Then get x, y, z values about the target in *Figure 4.3.5*:

$$x = \frac{b * (u_L - o_x)}{u_L - u_R} \quad (4.3.14)$$

$$y = \frac{b * f_x * (v_L - o_y)}{f_y * (u_L - u_R)} \quad (4.3.15)$$

$$z = \frac{b * f_x}{u_L - u_R} \quad (4.3.16)$$

The D435i also has an infrared (IR) projector that projects a pattern of dots into the scene. These dots serve as reference points for the depth calculations. It helps to achieve accurate depth measurements in various lighting conditions.

4.3.3 Test Results

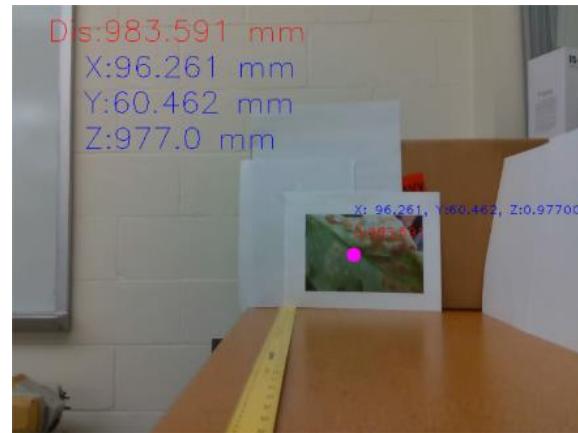


Figure 0.19

For Figure 4B.6, D435i is 0.25 meters away from the picture.

Distance estimated is 0.249307 meters.

Error is 0.0027%.



Figure 0.20

For Figure 4B.7, D435i is 0.5 meters away from the picture.

Distance estimated is 0.489262 meters.

Error is 2.1476%

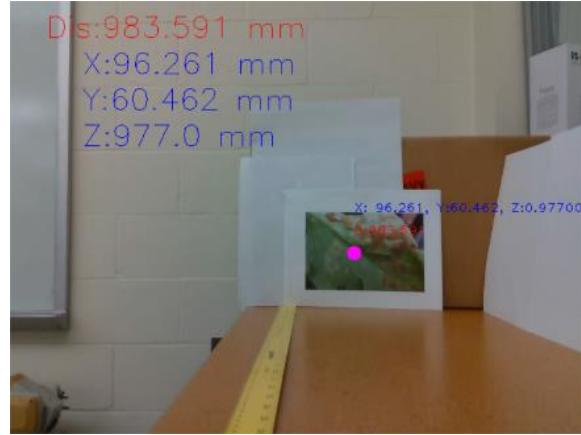


Figure 0.21

For Figure 4B.8, D435i is 1 meter away from the picture.

Distance estimated is 0.983591 meters.

Error is 1.6409%

The test result errors are less than 2cm, and errors are less than 5% which meet the requirements of the assumption.

4.4 Spray control system analysis

The physical problems of the spray system make it mainly necessary to analyze the nozzles and water pumps and determine the specifications according to customer needs.

It's required to get the appropriate spraying angle and spraying distance according to the spraying range required by the customer. It also needs to analyze the parameters required by the water pump by calculating the flow rate and pressure of the nozzle and simulating the working status of the nozzle.

The customer hopes that our final product can be 1 to 2 meters away from the nozzle target. The position can reach a spray range of up to 40 cm in diameter, which can basically cover the leaves of the target plants.

In actual spraying situations, the coverage of the nozzle is related to the spraying angle and spraying distance. Assuming that, under ideal circumstances with water, the angle remains unchanged during the entire spraying distance.

THEORETICAL SPRAY COVERAGE (AT VARIOUS DISTANCES FROM NOZZLE ORIFICE)

Spray Angle	2 in.	5 cm	4 in.	10 cm	6 in.	15 cm	8 in.	20 cm	10 in.	25 cm	12 in.	30 cm	15 in.	40 cm	18 in.	50 cm	24 in.	60 cm	30 in.	70 cm	36 in.	80 cm	48 in.	100 cm
5°	0.2	0.4	0.4	0.9	5	1.3	7	18	9	22	1.1	2.6	1.3	3.5	1.6	4.4	2.1	5.2	2.6	6.1	3.1	7.0	4.2	8.7
10°	0.4	0.9	0.7	1.8	1.1	2.6	1.4	3.5	1.8	4.4	2.1	5.3	2.6	7.0	3.1	8.8	4.2	10.5	5.2	12.3	6.3	14.0	8.4	17.5
15°	0.5	1.3	1.1	2.6	1.6	4.0	2.1	5.3	2.6	6.6	3.2	7.9	3.9	10.5	4.7	13.2	6.3	15.8	7.9	18.4	9.5	21.1	12.6	26.3
20°	0.7	1.8	1.4	3.5	2.1	5.3	2.8	7.1	3.5	8.8	4.2	10.6	5.3	14.1	6.4	17.6	8.5	21.2	10.6	24.7	12.7	28.2	16.9	35.3
25°	0.9	2.2	1.8	4.4	2.7	6.7	3.5	8.9	4.4	11.1	5.3	13.3	6.6	17.7	8.0	22.2	10.6	26.6	13.3	31.0	15.9	35.5	21.2	44.3
30°	1.1	2.7	2.1	5.4	3.2	8.0	4.3	10.7	5.4	13.4	6.4	16.1	8.1	21.4	9.7	26.8	12.8	32.2	16.1	37.5	19.3	42.9	25.7	53.6
35°	1.3	3.2	2.5	6.3	3.8	9.5	5.0	12.6	6.3	15.8	7.6	18.9	9.5	25.2	11.3	31.5	15.5	37.8	18.9	44.1	22.7	50.5	30.3	63.1
40°	1.5	3.6	2.9	7.3	4.4	10.9	5.8	14.6	7.3	18.2	8.7	21.8	10.9	29.1	13.1	36.4	17.5	43.7	21.8	51.0	26.2	58.2	34.9	72.8
45°	1.7	4.1	3.3	8.3	5.0	12.4	6.6	16.6	8.3	20.7	9.9	24.9	12.4	33.1	14.9	41.4	19.9	49.7	24.8	58.0	29.8	66.3	39.7	82.8
50°	1.9	4.7	3.7	9.3	5.6	14.0	7.5	18.7	9.3	23.3	11.2	28.0	14.0	37.3	16.8	46.6	22.4	56.0	28.0	65.3	33.6	74.6	44.8	93.3
55°	2.1	5.2	4.2	10.4	6.3	15.6	8.3	20.8	10.3	26.0	12.5	31.2	15.6	41.7	18.7	52.1	25.0	62.5	31.2	72.9	37.5	83.3	50.0	104
60°	2.3	5.8	4.6	11.6	6.9	17.3	9.2	23.1	11.5	28.9	13.8	34.6	17.3	46.2	20.6	57.7	27.7	69.3	34.6	80.8	41.6	92.4	55.4	115
65°	2.5	6.4	5.1	12.7	7.6	19.1	10.2	25.5	12.7	31.9	15.3	38.2	19.2	51.0	22.9	63.7	30.5	76.5	38.2	89.2	45.8	102	61.2	127
70°	2.8	7.0	5.6	14.0	8.4	21.0	11.2	28.0	14.0	35.0	16.8	42.0	21.0	56.0	25.2	70.0	33.6	84.0	42.0	98.0	50.4	112	67.2	140
75°	3.1	7.7	6.1	15.4	9.2	23.0	12.3	30.7	15.3	38.4	18.4	46.0	23.0	61.4	27.6	76.7	36.8	92.1	46.0	107	55.2	123	73.6	153
80°	3.4	8.4	6.7	16.8	10.1	25.2	13.4	33.6	16.8	42.0	20.2	50.4	25.2	67.1	30.3	83.9	40.3	101	50.4	118	60.4	134	80.6	168
85°	3.7	9.2	7.3	18.3	11.0	27.5	14.7	36.7	18.3	45.8	22.0	55.0	27.5	73.3	33.0	91.6	44.0	110	55.0	128	66.0	147	88.0	183
90°	4.0	10.0	8.0	20.0	12.0	30.0	16.0	40.0	20.0	50.0	24.0	60.0	30.0	80.0	36.0	100	48.0	120	60.0	140	72.0	160	96.0	200
95°	4.4	10.9	8.7	21.8	13.1	32.7	17.5	43.7	21.8	54.6	26.2	65.5	32.8	87.3	39.3	109	52.4	131	65.5	153	78.6	175	105	218
100°	4.8	11.9	9.5	23.8	14.3	35.8	19.1	47.7	23.8	59.6	28.6	71.5	35.8	95.3	43.0	119	57.2	143	71.6	167	85.9	191	114	238
110°	5.7	14.3	11.4	28.6	17.1	42.9	22.8	57.1	28.5	71.4	34.3	85.7	42.8	114	51.4	143	68.5	171	85.6	200	103	229	—	286
120°	6.9	17.3	13.9	34.6	20.8	52.0	27.7	69.3	34.6	86.6	41.6	104	52.0	139	62.4	173	83.2	208	104	243	—	—	—	—
130°	8.6	21.5	17.2	42.9	25.7	64.3	34.3	85.8	42.9	107	51.5	129	64.4	172	77.3	215	103	257	—	—	—	—	—	—
140°	10.9	27.5	21.9	55.0	32.9	82.4	43.8	110	54.8	137	65.7	165	82.2	220	98.6	275	—	—	—	—	—	—	—	—
150°	14.9	37.3	29.8	74.6	44.7	112	59.6	149	74.5	187	89.5	224	112	299	—	—	—	—	—	—	—	—	—	—
160°	22.7	56.7	45.4	113	68.0	—	170	90.6	227	113	284	—	—	—	—	—	—	—	—	—	—	—	—	—
170°	45.8	114	91.6	229	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Figure 0.22 [10]

The calculation formula is

$$\text{angle} = \text{Atan}\left(\frac{\text{Coverage}}{\text{Distance}}\right) \quad (4.4.1)$$

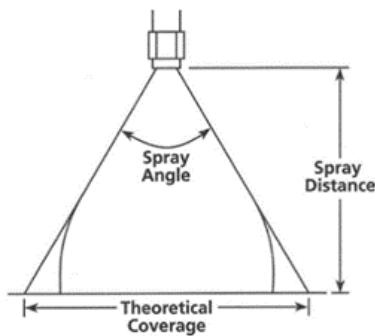


Figure 0.23 [10]

This *Figure 4.4.1* shows that a longer spraying distance will result in a smaller spraying angle. In this case, the ideal spraying angle needs to be controlled within 25° and the spraying distance should be more than one meter.

Determine the flow rate of the nozzle based on the pesticide spray volume, ground speed and coverage through the formula of the “5940” method.

$$gpm = \frac{gpa * mph * W}{5940} [11] \quad (4.4.2)$$

The amount of pesticides required for general crop leaves, such as tomato leaves, is 50-75 gallons per acre after dilution with water. The driving speed of agricultural vehicles connected to robotic arms is generally 0-5 mph. The flow rate required by the nozzle is about 0.3 - 0.8 gallons per minute with considering the 40 cm of theoretical coverage.

The *Figure 4.4.3* shows the nozzle flow rate under different nozzle pressures and different nozzle diameters.

Nozzle Size*	Orifice Dia.(In.)	40 PSI	100 PSI	250 PSI	500 PSI	600 PSI	700 PSI	800 PSI	1000 PSI	1200 PSI	1500 PSI	2000 PSI	2500 PSI	3000 PSI	3500 PSI	4000 PSI	5000 PSI	6000 PSI	7000 PSI
2	.034	.20	.32	.50	.71	.77	.80	.89	1.0	1.1	1.2	1.4	1.6	1.7	1.9	2.0	2.24	?	?
4	.052	.40	.63	1.00	1.40	1.60	1.70	1.80	2.0	2.2	2.5	2.8	3.1	3.5	3.8	4.0	4.5	4.9	5.3
4.5	.055	.45	.71	1.10	1.50	1.70	1.90	2.00	2.2	2.4	2.8	3.0	3.6	3.9	4.3	4.5	5.0	5.5	5.9
5	.057	.50	.79	1.30	1.80	1.90	2.10	2.20	2.5	2.8	3.1	3.6	4.0	4.4	4.7	5.0	5.6	6.1	6.6
5.5	.060	.55	.87	1.40	1.90	2.10	2.30	2.50	2.8	3.0	3.4	3.8	4.4	4.8	5.2	5.5	6.2	6.7	7.3
6	.062	.60	.95	1.50	2.10	2.30	2.50	2.70	3.0	3.2	3.7	4.2	4.8	5.2	5.6	6.0	6.7	7.3	7.9
6.5	.064	.65	1.00	1.70	2.30	2.50	2.70	2.90	3.3	3.6	4.0	4.6	5.2	5.7	6.0	6.5	7.3	8.0	8.6
7	.067	.70	1.10	1.80	2.50	2.70	2.90	3.10	3.5	3.8	4.3	5.0	5.6	6.1	6.6	7.0	7.8	8.6	9.3
7.5	.070	.75	1.20	1.90	2.70	2.90	3.20	3.40	3.8	4.1	4.6	5.3	6.0	6.5	7.0	7.5	8.4	9.2	9.9
8	.072	.80	1.30	2.00	2.80	3.10	3.40	3.60	4.0	4.4	5.0	5.6	6.2	7.0	7.5	8.0	8.9	9.8	10.6
8.5	.074	.85	1.30	2.20	3.00	3.30	3.60	3.80	4.3	4.6	5.3	6.0	6.7	7.4	8.0	8.5	9.5	10.4	11.2
9	.076	.90	1.40	2.30	3.20	3.50	3.80	4.00	4.5	5.0	5.5	6.4	7.1	7.8	8.5	9.0	10.1	11.0	11.9
9.5	.078	.95	1.50	2.40	3.40	3.70	4.00	4.30	4.8	5.2	5.8	6.8	7.6	8.3	9.0	9.5	10.62	11.6	12.6
10	.080	1.00	1.60	2.50	3.50	3.90	4.20	4.50	5.0	5.4	6.1	7.0	8.0	8.7	9.4	10.0	11.2	12.2	13.2
12	.087	1.20	1.90	3.00	4.20	4.60	5.00	5.40	6.0	6.4	7.3	8.4	9.5	10.4	11.2	12.0	13.42	14.6	15.70
12.5	.089	1.25	1.98	3.13	4.42	4.84	5.23	5.59	6.25	6.85	7.65	8.84	9.88	10.83	11.69	12.50	13.98	15.3	16.5
13	.091	1.30	2.06	3.25	4.60	5.03	5.44	5.81	6.50	7.12	7.96	9.19	10.28	11.26	12.16	13.00	14.53	15.8	17.02
15	.094	1.50	2.40	3.80	5.30	5.80	6.40	6.80	7.5	8.2	9.2	10.6	12.0	12.9	14.0	15.00	16.8	18.4	19.8
20	.109	2.00	3.20	5.00	7.10	7.80	8.40	9.00	10.0	10.8	12.2	14.2	16.0	17.4	18.8	20.00	22.36	24.3	28
25	.125	2.50	3.95	6.25	8.84	9.68	10.46	11.18	12.50	13.69	15.31	17.68	19.76	21.65	23.59	25.00	27.95	30.4	35
30	.141	3.00	4.70	7.50	10.60	11.60	12.80	13.60	15.0	16.40	18.40	21.2	24.0	26.0	28.0	30.00	33.54	36.6	42
40	.156	4.00	6.30	10.00	14.20	15.60	16.80	18.00	20.0	21.60	24.40	28.4	32.0	34.8	37.6	40.00	44.72	48.75	56
50	.172	5.00	7.91	12.50	17.68	19.36	20.92	22.36	25.00	27.39	30.62	35.36	39.53	43.30	46.77	50.00	55.90	60.93	70
60	.188	6.00	9.49	15.00	21.21	23.24	25.10	26.83	30.00	32.86	36.74	42.43	47.43	51.96	56.12	60.00	67.08	73.11	84

Figure 0.24

The formula used is the Torricelli equation to calculate the relationship between pressure, nozzle diameter and nozzle flow rate.

$$Q = \mu * A * (2 P / \rho)^{0.5} \quad (4.4.3)$$

There is no need to consider the spray angle here since it has no effect on the flow rate of the nozzle.

It can be seen from the upper left corner of the *Figure 4.4.3* that when the flow rate is 0.2-0.5gpm, the pressure is 40psi (0.27Mpa), and the nozzle diameter changes from 0.8mm to 1.4 mm, which is roughly within the range of general agricultural nozzles, so the pressure at the nozzle should be around 0.27Mpa

A theoretical model of a nozzle by CAD. The diameter of the nozzle was 1.5mm. We used ASYS Fluent to simulate the nozzle.

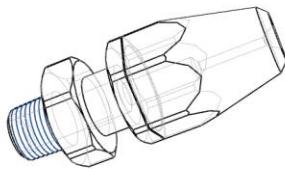


Figure 0.20 CAD mode

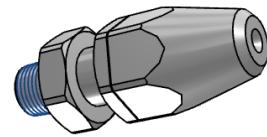


Figure 0.19 CAD model

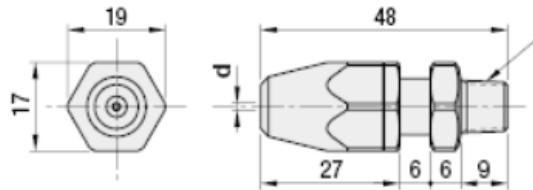


Figure 0.21

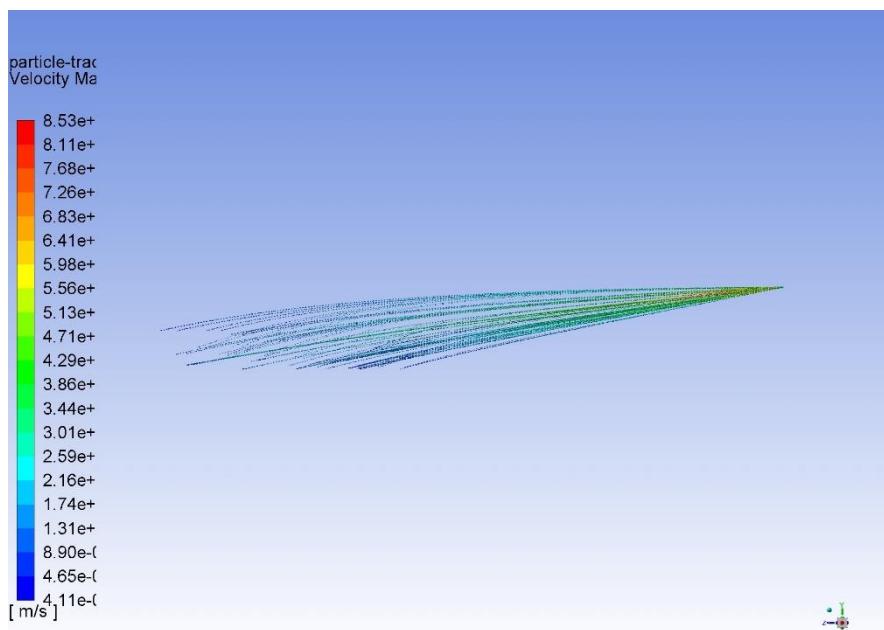


Figure 0.22 ASYS Fluent

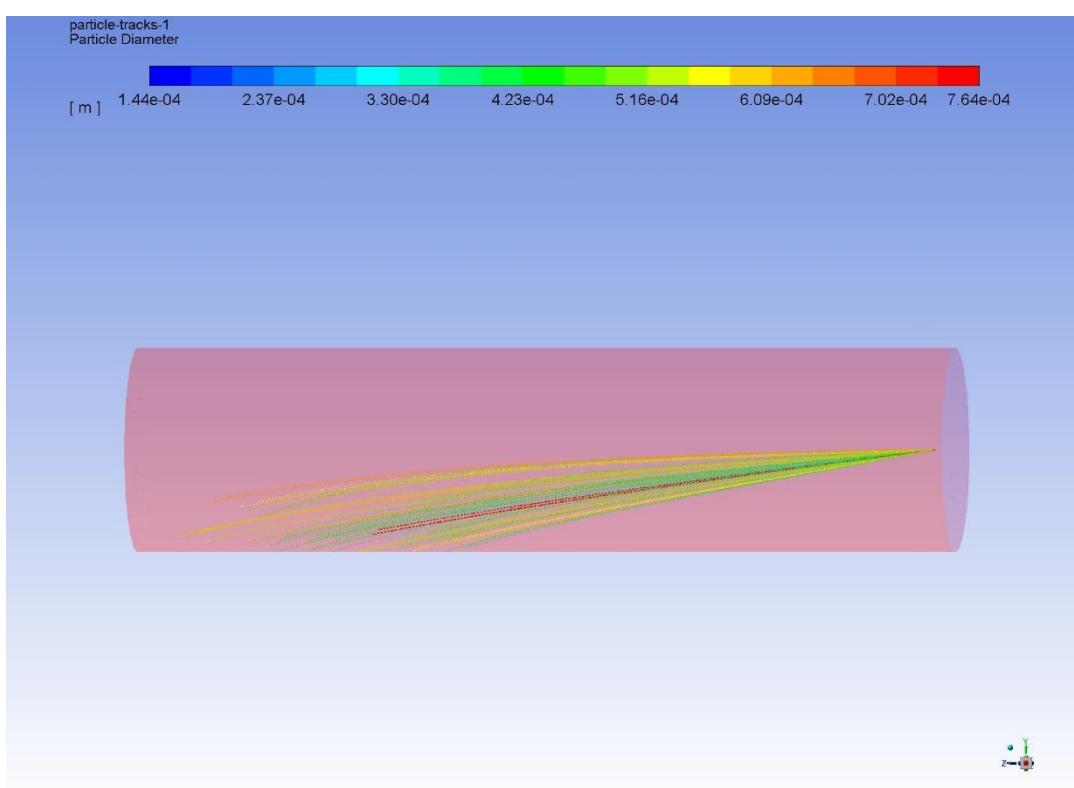


Figure 0.29 ASYS Fluent

The spray angle is set to 20°, the internal pressure of the nozzle is set to 0.5 MPa and the flow rate is 1.9 LPM.

Through simulation, the spraying distance of the nozzle can reach 3-4 meters under the condition of such high pressure and high flow rate, which has exceeded the 1-2 meters spraying distance required by the customers. However, in actual spraying situations, the pressure is not likely to exceed 0.5Mpa, so the spraying distance will be controlled at about 1-2m, which can meet the requirement from customers.

The ideal pressure inside the nozzle has been obtained, which directly determines the pressure provided by the water pump, and the pressure at the nozzle is provided by the water pump. However, it is worth considering that the friction coefficient of the water pipe itself and the bending of the mechanical arm will cause the loss of pressure provided by the water pump, so the pressure provided by the water pump must be greater than the calculated value of the pressure inside the nozzle.

In fluid mechanics, the pressure drop produced by the water pump to the nozzle can be calculated. Firstly, calculate the flow rate of fluid in the pipe:

$$\dot{M} = \rho * u * A [13] \quad (4.4.4)$$

There are many diameters of agricultural water pipes, the most regular sizes are 6-7mm in diameter, which can determine the cross-section(A) in the equation, so the fluid velocity(\dot{M}) is roughly 0.57-1.5 m/s.

Determine the state of fluid in the pipeline based on Reynolds number:

$$Re = \frac{u * D * \rho}{\mu} > 2000 [13] \quad (4.4.5)$$

When the Reynolds number is greater than 2000, the fluid is turbulent. Correspondingly, the fluid is laminar flow when the Reynolds number is less than 2000. The calculation methods of the pressure are different due to different states of the fluid.

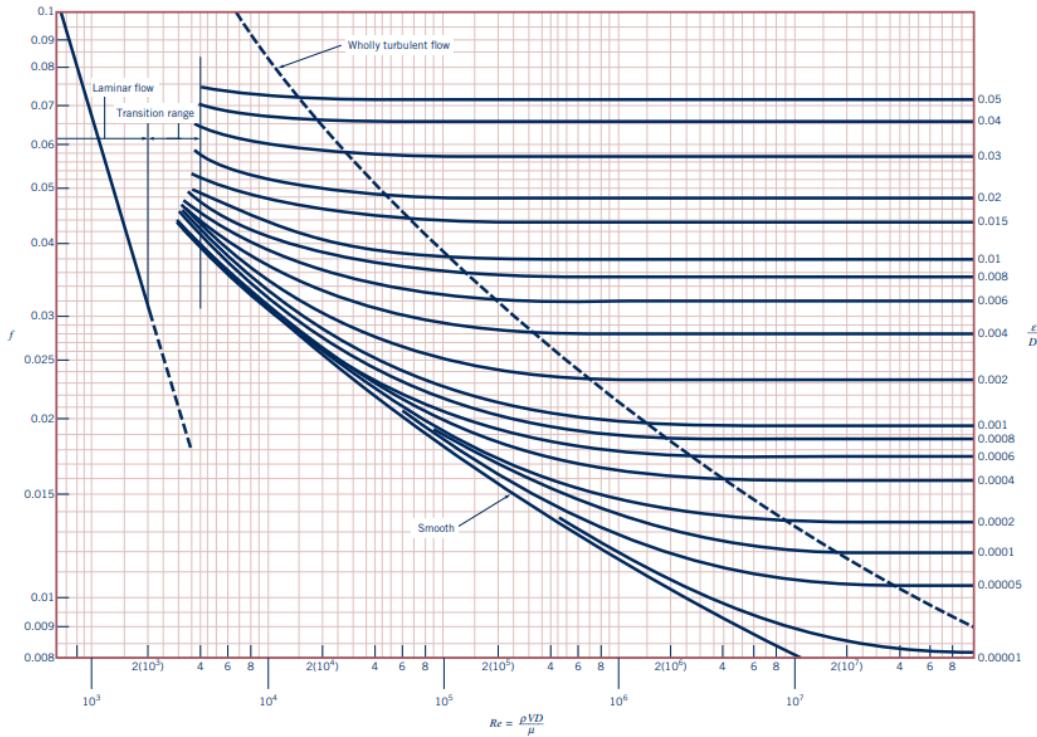


Figure 0.30 Moody chart

The following equation from Colebrook is valid for the entire nonlaminar range of the Moody chart:

$$\frac{1}{\sqrt{f}} = -2.0 \log \left(\frac{\varepsilon/D}{3.7} + \frac{2.51}{Re\sqrt{f}} \right) [13] \quad (4.4.6)$$

For turbulent flow in smooth pipes ($\varepsilon/D = 0$) with $Re < 10^5$ the equation can be simplified as:

$$f = \frac{0.316}{Re^{1/4}} [13] \quad (4.4.7)$$

The pressure drop can be calculated by the following formula:

$$\Delta P = f \frac{l}{D} \frac{1}{2} \rho V^2 = 270 \text{ pa} - 632 \text{ pa} [13] \quad (4.4.8)$$

Considering that the bending situation of water pipes is complicated, it is necessary to simplify the bending situation of water pipes. Assuming that the bending is exactly 90° , the loss coefficient is:

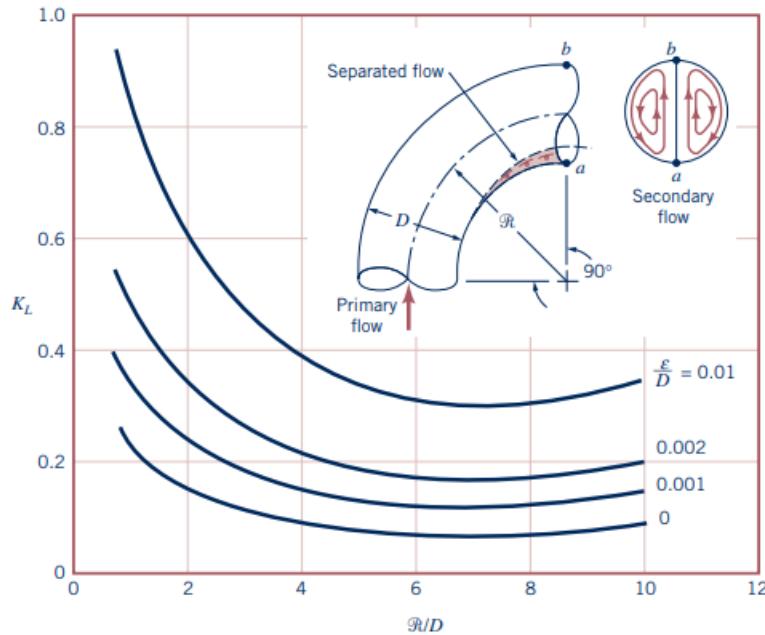


Figure 0.31

Equation of loss coefficient:

$$K_L = \frac{\Delta P_m}{\frac{1}{2} \rho V^2} \quad (4.4.9)$$

$$\Delta P_{final} = \Delta P + \Delta P_m = 302.5 - 664.5 \text{ Pa} \quad (4.4.10)$$

The final pressure difference result is very small since it is related to the length and diameter of the water pipe, the fluid velocity, density, and viscosity. The length and diameter of the water pipe used in this project are all small, so it is nearly negligible.

Pressure and flow rate are two important specifications of the water pump. Generally, manufacturers will display the performance curve of the water pump in the datasheet, that is, the curve of pressure and flow rate. So, the required pump in the future should consider both the pressure that must be calculated and the performance curve from the datasheet.

Products may choose:

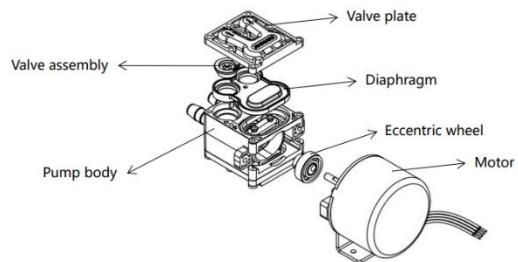


Figure 0.27

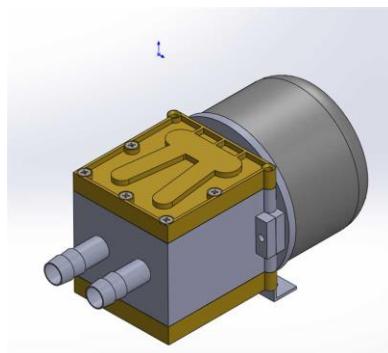


Figure 0.26

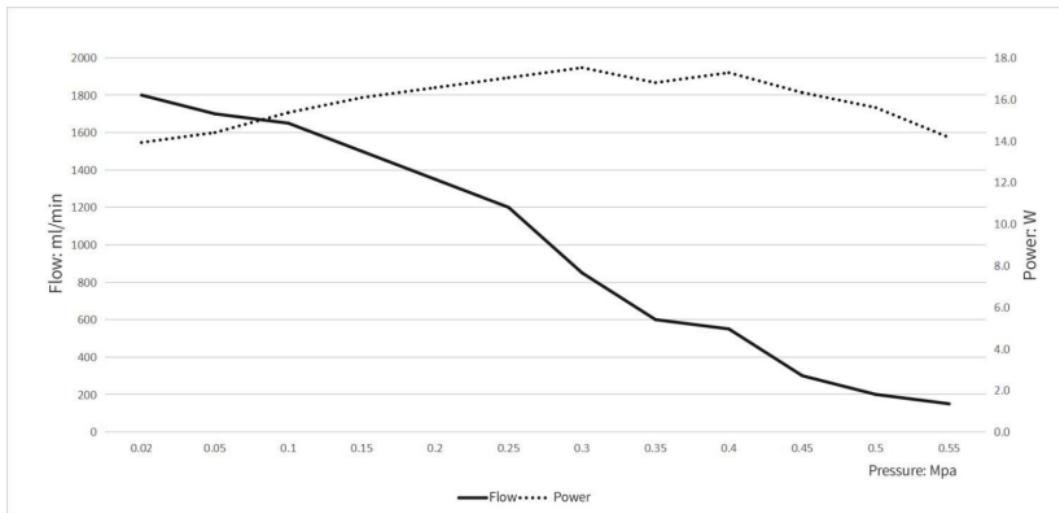


Figure 0.34 Testing curve

This product produces a flow rate of 1-1.4 liters per minute under a pressure of 0.1-0.27Mpa, which can meet the design requirements.

The analysis of the spray system is very helpful for the design of the project, because this basically determines the specifications of the nozzle and water pump, what is more, the spray system directly determines the precise amount of pesticides for future products.

In the future, in order to obtain accurate spraying volume and spraying distance, it is necessary to program the water pump according to actual conditions such as the distance between the nozzle and the plant, the flow rate of the water pump, the power of the water pump and the working time to control the precise spraying of the water pump.

5 Alpha Prototype

5.1 Alpha Prototype Construction

During the Alpha prototype phase, we focused on the robotic arm and spray system. We want readers to understand how the robotic arm and spray system work independently, as well as their individual testing. For the robotic arm, we need to verify the accuracy of its movements controlled by code through testing. For the spray system, we need to find the appropriate distance between the nozzle and the plants to ensure the rational use of pesticides. Then, the nozzle of the spray system should be attached to the end of the robotic arm.

In the construction of the Alpha prototype, a challenge we encountered with the robotic arm was its limited degrees of freedom. Although a four-degree-of-freedom arm was sufficient for our design, it could not reach some specified positions. Although we could mitigate this issue by modifying the code, it still posed some challenges to our Alpha prototype construction. In building the spray system, excessive water pressure could cause the hose connections to burst, potentially leading to short circuits and corrosion of the circuits.

Since we planned to add a camera and some electronic components to the Alpha prototype later, we temporarily mounted the robotic arm and spray system on a rectangular wooden board, which facilitates improvements during the construction of the Beta prototype.

5.2 Alpha Prototype Testing

The purpose of testing the robotic arm is to determine the accuracy with which it reaches specified positions. This is not merely a test of the precision of the robotic arm itself, but rather a test of the accuracy of the code we have written to control its movements. The logic of the code we wrote may lead to deviations in the movement of the robotic arm. We need to measure this deviation to ascertain whether it falls within our anticipated margin of error.

For the spray system, simulations are performed to record the spray coverage at different distances and find a spraying distance that can completely cover the spray range of the target plants. And we will verify the accuracy and rationality of the simulation through the actual measured coverage, and finally establish the distance between the nozzle and the plant to ensure that this distance can maximize pesticide utilization and reduce pesticide waste.

5.3 Alpha prototype build status

Table 0.1 Alpha prototype build status

Component	Percentage complete	Completed date
Robotic Arm Assembly	100%	2/4/2024
Spray Nozzles Installation	100%	2/4/2024
Water pump and pipes installation	100%	2/4/2024
Circuit connection	100%	2/5/2024
Battery and circuit integration	100%	2/5/2024

For the completeness of subsequent testing, the robotic arm and the spraying system need to be installed together. In the Alpha prototype build status, we installed the system on the wooden board one after another, we fixed the robotic arm to one end of the wooden board with screws, and then we installed the water pump in the middle of the wooden board, connect the water outlet of the water pump to the water pipe and nozzle, and fix the nozzle at the end effector of the robotic arm. On the

other end of the template, we install the Arduino Mega control board and breadboard to design and connect the circuit.

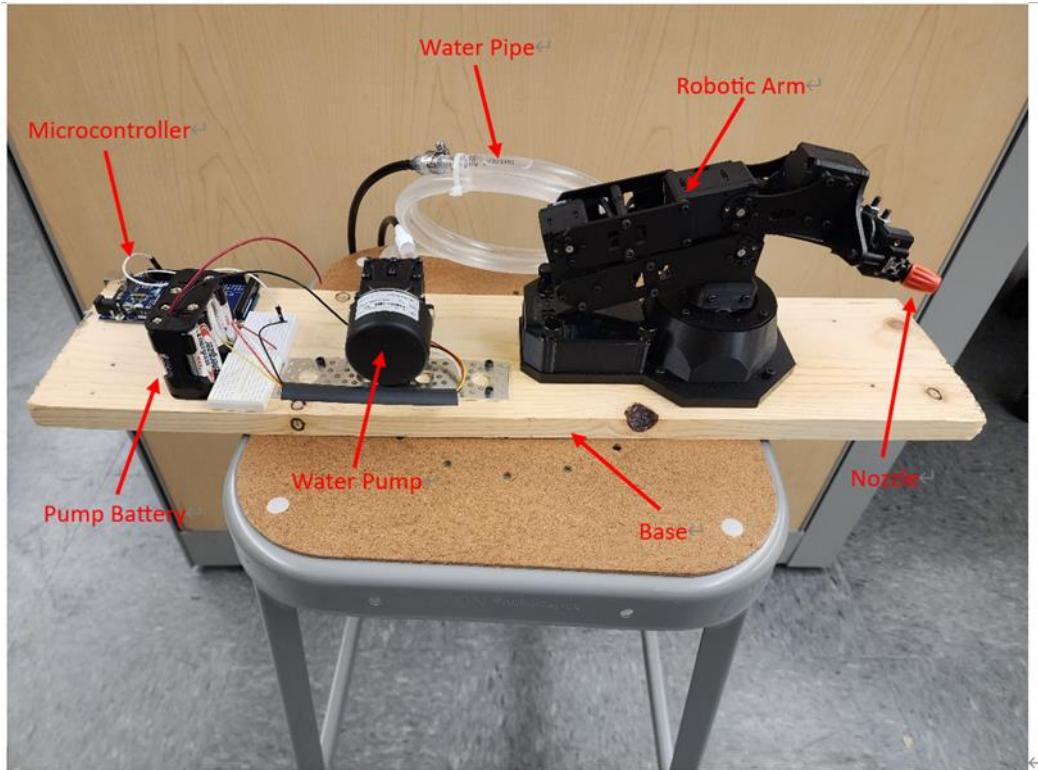


Figure 0.1 Alpha prototype

5.4 Experiment design and testing plan status for Alpha prototype

5.4.1 Experiment design

Spray System

The spray system mainly consists of two parts: a pump and a nozzle. In the first experiment, in order to ensure that the nozzle of the robot arm end effector maintains an optimal distance from the plant and covers the entire plant to prevent disease, we will determine the most effective distance in the spray through actual spraying, and calculate the simulated spraying. The relative error obtained by comparing the coverage area with the actual spray coverage area.

Robotic Arm

To test the accuracy of a robotic arm's movement controlled by code, we first need to input the coordinates of the potential positions the robotic arm might reach into the code. Next, we run the code to move the robotic arm and measure the discrepancy between the actual position reached by the robotic arm and the input position. Finally, we assess whether this actual deviation falls within our anticipated margin of error.

5.4.2 Testing plan status

Table 50.2.1

Test	Percentage complete	Complete date
Position Control Test	100%	2/4/2024
Accuracy of spraying distance	100%	2/8/2024

Based on the table above, the alpha test has been 100% completed.

5.5 Test Results

5.5.1 Spray system

We built a nozzle model, and we used Ansys Fluent to simulate the spraying effect of the nozzle. We set the flow rate of the nozzle outlet to 0.5L/min, and the water pressure to 2-3 bar.

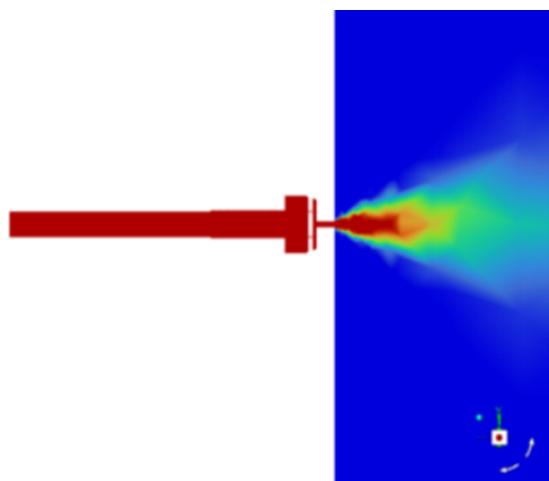


Figure 5.5.1.1

To determine spray coverage, we performed spray simulations and recorded nozzle coverage
Table 5.5.1.1

every 10 cm. The recorded data in the following *Table 5.5.1.1*:

Distance	10cm	20cm	30cm	40cm	50cm	60cm
Coverage (cm ²)	15.41	61.678	138.78	246.71	385.48	555.1

In the actual test, we first measure the spraying distance of the nozzle:



Figure 5.5.1.2

We have tested that the maximum spraying distance can be over to 150cm, and the water mist first wets the first half of the distance during the spraying process. The optimal effective range of the spray is about 40cm-60cm.

Because environmental factors such as gravity are taken into account in the actual measurement, we placed the nozzle vertically downward and moved the nozzle to maintain a distance from 10cm to 60cm from the ground, and measured the spray range every ten centimeters.

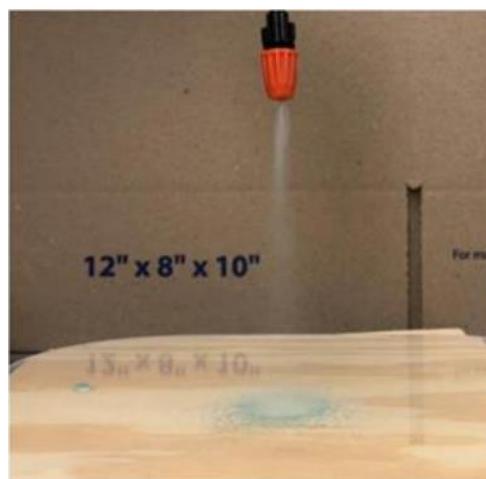


Figure 5.5.1.3

We get the actual coverage data and compare it with the data from the simulation and calculate the relative error.

Table 5.5.1.2

Distance	10cm	20cm	30cm	40cm	50cm	60cm
Simulated Coverage (cm ²)	15.41	61.678	138.78	246.71	385.48	555.1
Real Coverage (cm ²)	17.625	63.59	132.665	254.34	314	530.66
Relative error	14.37%	3.09%	4.4%	3.09%	18.54%	4.5%

The theoretical calculation formula of this model is:

$$\text{angle} = \text{Atan}\left(\frac{\text{Coverage}}{\text{Distance}}\right) \quad (5.5.1.1)$$

Finally, it can be verified through simulation and actual measurement results that the spraying distance between plants and nozzles needs to be controlled at 40-60cm to achieve the best spraying effect and reduce pesticide loss.

5.5.2 Robotic arm

In testing the robotic arm, we initially conduct a large number of tests to determine the positions the arm is most likely to reach. We then transmit these positions to the robotic arm via code. After waiting for the arm to move, we calculate the discrepancies between the actual position of the

Table 5.5.2.1

arm and the input positions. These discrepancies include errors on the x, y, and z axes, as well as the overall error. This process helps us evaluate the precision of the arm's movements and the effectiveness of the control code in achieving the desired positions.

Number	Target position (m)			Actual position (m)			Error (m)			Total error (m)
	X	Y	Z	X	Y	Z	X	Y	Z	
1	0.1	0.1	0.1	0.12	0.12	0.1	0.02	0.02	0	0.02
2	0.04	0.17	0.24	0.05	0.22	0.24	0.01	0.05	0	0.03

3	0	0.34	0.2	0	0.24	0.21	0	0.1	0.01	0.08
4	-0.05	-0.2	0.3	-0.06	-0.24	0.19	0.01	0.04	0.11	0.05
5	0.23	0.23	0.23	0.18	0.18	0.19	0.05	0.05	0.04	0.08
6	0.04	0.05	0.29	0.08	0.1	0.29	0.04	0.05	0	0.02
7	0.03	0.03	0.29	0.07	0.07	0.29	0.04	0.04	0	0.01
8	0.04	0.03	0.29	0.07	0.05	0.29	0.03	0.02	0	0
9	0.01	0.04	0.3	0	0.12	0.3	0.01	0.08	0	0.02
10	-0.09	0.02	0.29	-0.09	0.03	0.29	0	0.01	0	0

To facilitate data analysis and more visually demonstrate the discrepancies between the actual and target positions, I have created three charts. Each chart represents the comparison of errors between the actual and target positions along the x-axis, y-axis, and z-axis, respectively. These charts will help in understanding the precision of the robotic arm's movements on each axis individually and make it easier to identify patterns or specific areas where improvements may be necessary.

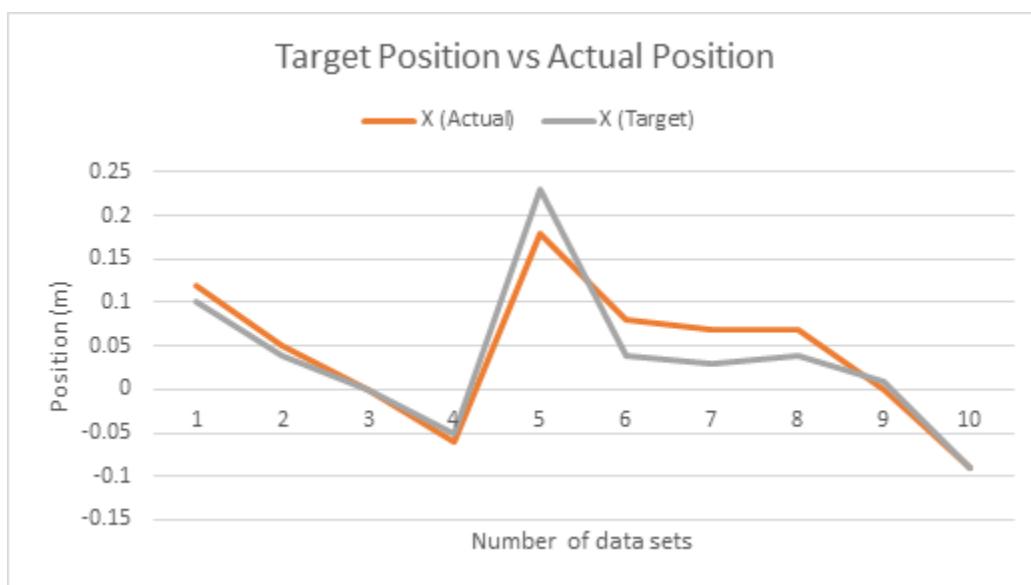


Figure 5.5.2.1

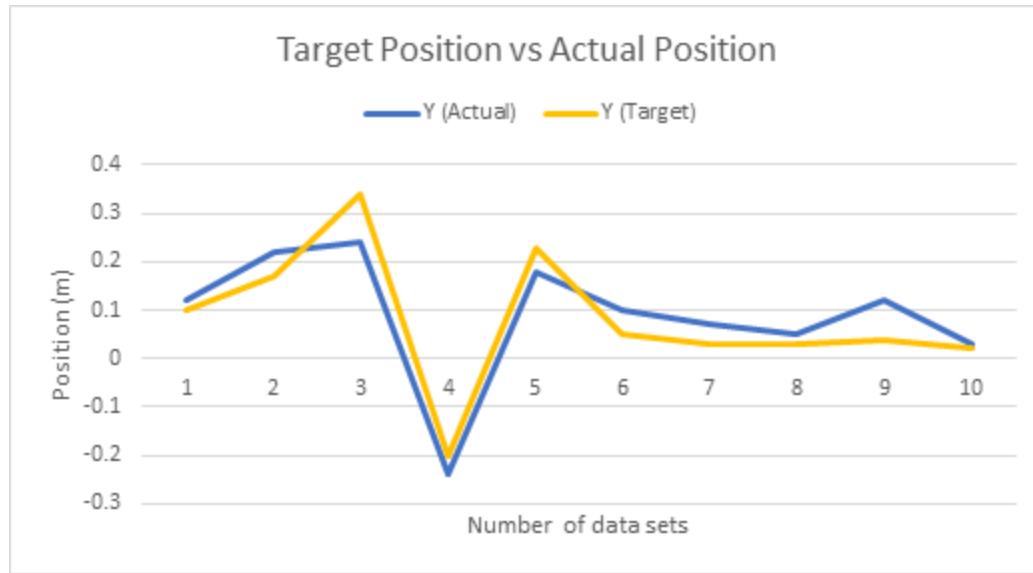


Figure 5.5.2.2

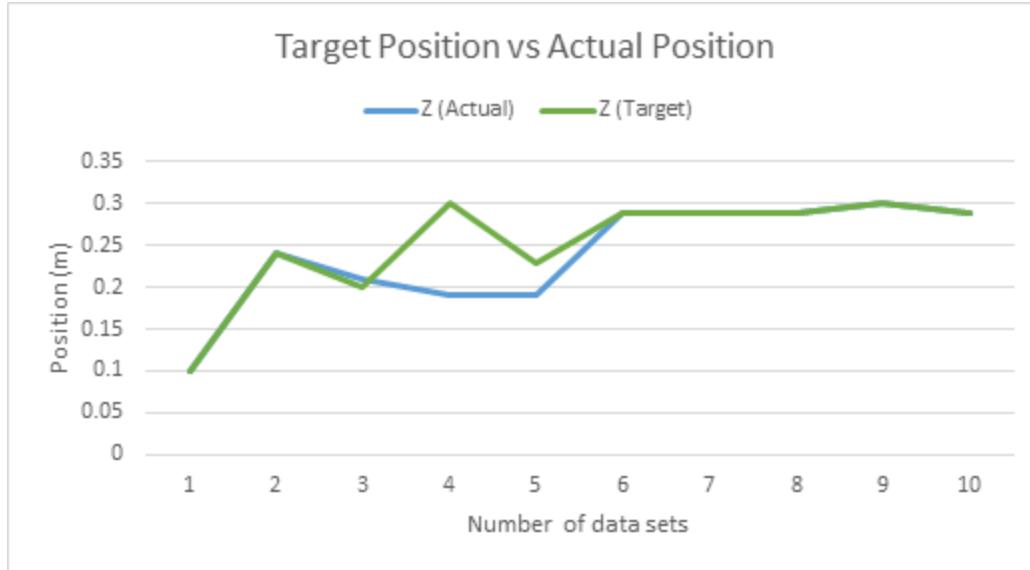


Figure 5.5.2.3

From the three charts mentioned, it can be observed that the error on the z-axis is the smallest.

Although the errors on the x-axis and y-axis are larger compared to the z-axis, they are relatively stable. This aligns with our expectations, indicating that the robotic arm performs with consistent accuracy in lateral movements, and has more precision in vertical movements. This stability and pattern of errors are essential for making further adjustments and improvements in the robotic arm's programming and mechanical design.

6 Beta Prototype

6.1 Beta Prototype Construction

In the preceding Alpha Prototype phase, we successfully achieved control over the robot arm movement and the water pump functionality, alongside ongoing efforts in refining the image recognition model. Moreover, we assembled some necessary components.

Transitioning to the Beta Prototype stage, significant progress has been made. We have completed two different image recognition models: the lesion prevention mode and the lesion treatment mode, and the successful integration of circuitry with the battery. All necessary assembly

tasks have been completed. In addition, we have also made improvements and optimizations based on Alpha Prototype. Considering the safety of the product, we have done waterproofing work to prevent water from dripping onto the circuit board and causing leakage and other problems. This marks a substantial milestone in our project's development. At the same time, we face many challenges. For the camera, we need to build our own operating environment and customize training models. We also need to combine image recognition functions and distance measurement functions. For information transmission, we need to implement information transmission between the robotic arm and the camera and signal transmission between the robotic arm and the spraying system.

6.2 Beta Prototype Testing

The test objectives for the camera system encompass two key aspects: image recognition and position measurement. Firstly, we aim to determine the capability of the image recognition model in correctly identifying crops or leaf lesions. Concurrently, we also evaluate the accuracy of position measurements and area measurements. This evaluation ensures that the position data collected meets the requirements of the robotic arm, while the area measurement matches the spray distance required for the nozzle to operate. And the resolution of the camera should reach 640×480 pixels.

For spraying system, by measuring the relationship between the nozzle's theoretical coverage and the distance between the nozzle and the plant, we hope to control the distance between the end effector of the robotic arm and the plant based on this test to ensure that the distance between the nozzle and the plant is maintained at a position where completely cover the whole plant. By designing the PID controller and building a feedback system, we can make the pesticide reach a stable target flow rate in a short time, thereby making the use of pesticides more efficient.

There is only one main objective in testing the robotic arm, which is the accuracy of the robotic arm movement. It is difficult to directly measure the accuracy of the robotic arm movement, so we

measure the actual arrival position of the end effector and compare it with the expected spraying position, analyze the error, and obtain the accuracy of the robotic arm.

6.3 Beta prototype build status

Table 6.3.1 Beta prototype build status

Component	Percentage complete	Completed Date
Robotic Arm Assembly	100%	3/8/2024
Spray Nozzles Installation	100%	3/8/2024
Water pump and pipes installation	100%	3/8/2024
Circuit connection	100%	3/9/2024
Battery and circuit integration	100%	3/11/2024
Camera installation	100%	3/11/2024

In the previous alpha prototype build, to easily modify the circuit, we assembled the robotic arm, camera, nozzle, motor, and pump set together. Now in the beta prototype, since the circuit is complete and the necessary components are installed, we improve on the previous build. First, we built a 17" × 11" × 7" box with thin wooden boards. We dug holes around it to allow the data cables to extend. We added a water barrier in the middle of the box. The water pump and water pipes are installed on one side of the water barrier, and the mobile power supply, Arduino board, and breadboard are installed on the other side. This avoids product safety issues caused by water leakage. And install the robotic arm, step motor and camera on the top of the box. Mount the LCD on the side of the box.

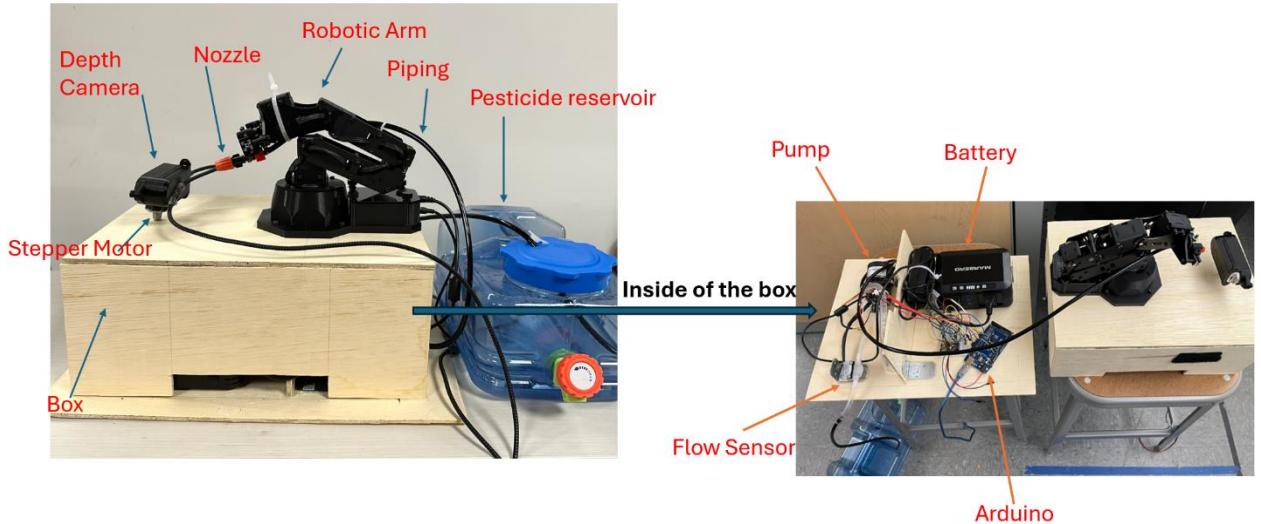


Figure 6.3.1 Beta Prototype

6.4 Experiment design and testing plan status for beta prototype

6.4.1 Experiment design

Spray System

The spray system mainly consists of two parts: water pump and nozzle. In the first experiment, to ensure that the nozzle of the robot arm's end effector maintains an optimal distance from the plant and covers the entire plant to prevent disease, we compare the calculated theoretical coverage rate with the actual coverage rate and calculate the relative error, and then we finally control the movement of the robotic arm based on the crop dimensions provided by the depth camera. The second experiment was to prevent pesticide waste by designing a PID controller and establishing a feedback system through a flow sensor to enable the water pump to maintain a stable flow rate.

Robotic Arm

There are two sets of tests. The first set of tests involves placing a target on the left and right sides of our product respectively, while the second set involves placing two targets on each side of the product to simulate a multi-target scenario. The purpose of both sets of tests is to calculate the accuracy of the robotic arm's movement, specifically whether the final position of the end effector matches the calculated spray position. If they do not match, we calculate the error value. To facilitate testing, we

use some colored blocks as targets. The reason for not using real plants as targets in the tests is because the image recognition model for detecting whole plant still needs improvement. Once the improvements are completed, we will replace the current image recognition model with the improved model and conduct the tests again.

Camera

The camera test is mainly divided into three parts, the image recognition accuracy test, the area accuracy test, and the position measurement accuracy test.

First, for the image recognition part, we will use the potted crops as test objects to verify the model. In this part, two models will be tested. Model-1 (Lesion Identification Model) is needed to test whether it can correctly identify leaf lesions, and Model-2 (Crop Identification Model) can correctly identify the entire crop.

The second part of the test for area accuracy will test the errors between the measured area and the actual area. We will continue using a potted crop, paint some spots on the leaves to simulate leaf lesions, and then measure their area (using the Lesion Identification Model). We will also use (Crop Identification Model) to measure the crop's overall area. Then we will compare the measured area with the actual area to calculate the error and confirm its performance.

In the third part, for the test of position measurement accuracy, we fix the camera at a known coordinate point and then move the object to a specific point within the coordinate system to determine the actual coordinates and measure its coordinates through the camera for comparison. Then we can compare the measured coordinates with actual coordinates to test its working performance.

6.4.2 Testing plan status

Table 6.4.2 Beta Test Status

Test	Percentage complete	Completed Date
Accuracy of spraying distance	100%	4/11/2024
PID controlling test	100%	4/10/2024
Coordinate conversion test	100%	3/15/2024

Camera transmission coordinate test	100%	3/16/2024
One target test	100%	3/18/2024
Two targets test	100%	3/20/2024
Replacement of improved recognition model test	100%	4/3/2024
Position measurement accuracy test	100%	3/24/2024
Image recognition accuracy test	100%	3/25/2024
Area accuracy test	100%	3/26/2024

Based on the table above, the beta test has been 100% completed now.

6.5 Test results

6.5.1 Spray system

Volume Test

There is no equipment for automatically adding pesticides in our system, so we need to monitor the number of pesticides used in real time and tell the user by displaying this value. We calculated the amount of pesticide through the flow sensor and displayed it on the LCD screen. Then to verify its accuracy, we conducted actual tests on the spray volume.

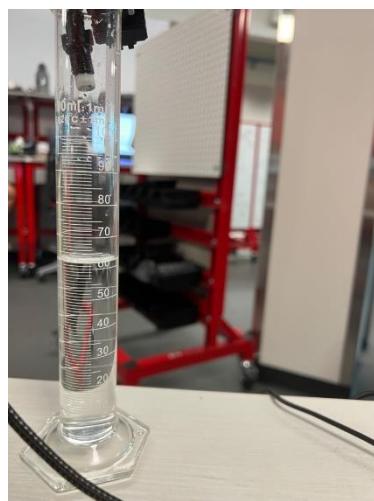


Figure 6.5.1.1.1

We controlled the spraying from one second to five seconds and recorded the actual spray volume by using a graduated cylinder. We recorded five sets of data, and then compared them with the values on the LCD screen to calculate the relative error.

Table 6.5.1.1.1

1s	1	2	3	4	5
measured/L	0.021	0.023	0.02	0.021	0.023
actual/L	0.022	0.025	0.022	0.022	0.024
relative error	4.55%	8.00%	9.09%	4.55%	4.17%
2s					
measured/L	0.041	0.041	0.039	0.04	0.039
actual/L	0.042	0.042	0.043	0.042	0.041
relative error	2.38%	2.38%	9.30%	4.76%	4.88%
3s					
measured/L	0.058	0.057	0.059	0.058	0.058
actual/L	0.062	0.06	0.062	0.062	0.061
relative error	6.45%	5.00%	4.84%	6.45%	4.92%
4s					
measured/L	0.077	0.074	0.075	0.079	0.079
actual/L	0.082	0.081	0.081	0.084	0.08
relative error	6.10%	8.64%	7.41%	5.95%	1.25%
5s					
measured/L	0.1	0.099	0.104	0.099	0.092
actual/L	0.109	0.101	0.106	0.098	0.096
relative error	8.26%	1.98%	1.89%	1.02%	4.17%

We plotted the error for each set of times.

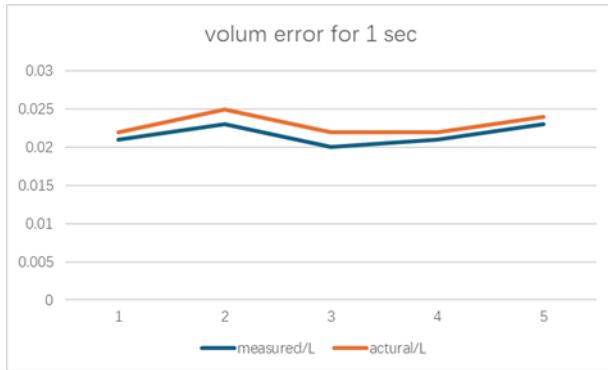


Figure 6.5.1.1.2

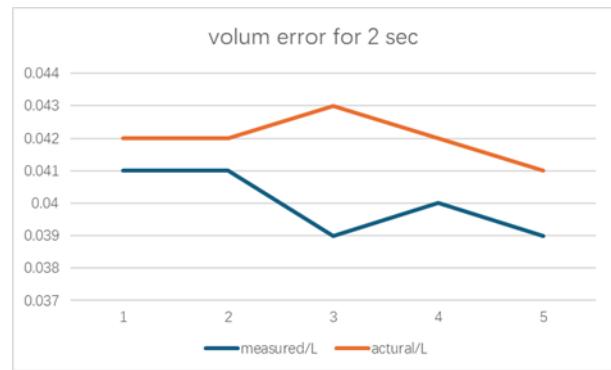


Figure 6.5.1.1.3



Figure 6.5.1.1.4

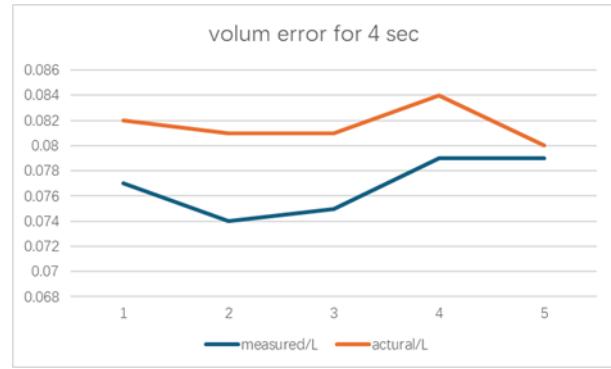


Figure 6.5.1.1.5

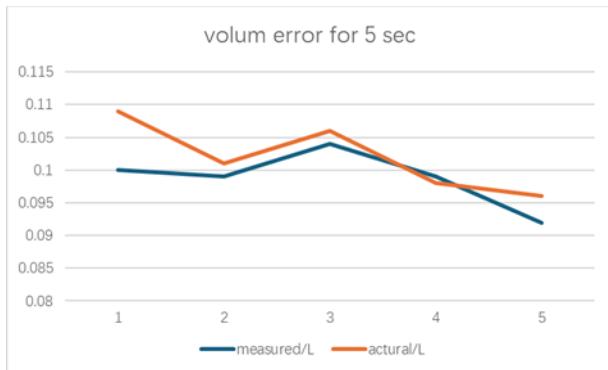


Figure 6.5.1.1.6

Flow rate control

During the control process, the instability of the water pump leads to unstable flow, which will cause the waste of pesticide to a certain extent. We hope to design a feedback system with a PID controller to achieve a stable flow output.

Before using PID, there was no feedback system established for the spray system. We connected the water pump with an external 12v power supply to the Arduino Mega 2560 and placed a flow sensor on the water pipe, which is connected to the water pump. We used the sensor readings to detect the flow output of the water pump at different speeds in real time.

We tested three flow rate changes with outputting 40%, 60%, and 100% duty cycle of the PWM signal respectively.

By designing a PID controller and building a feedback system:

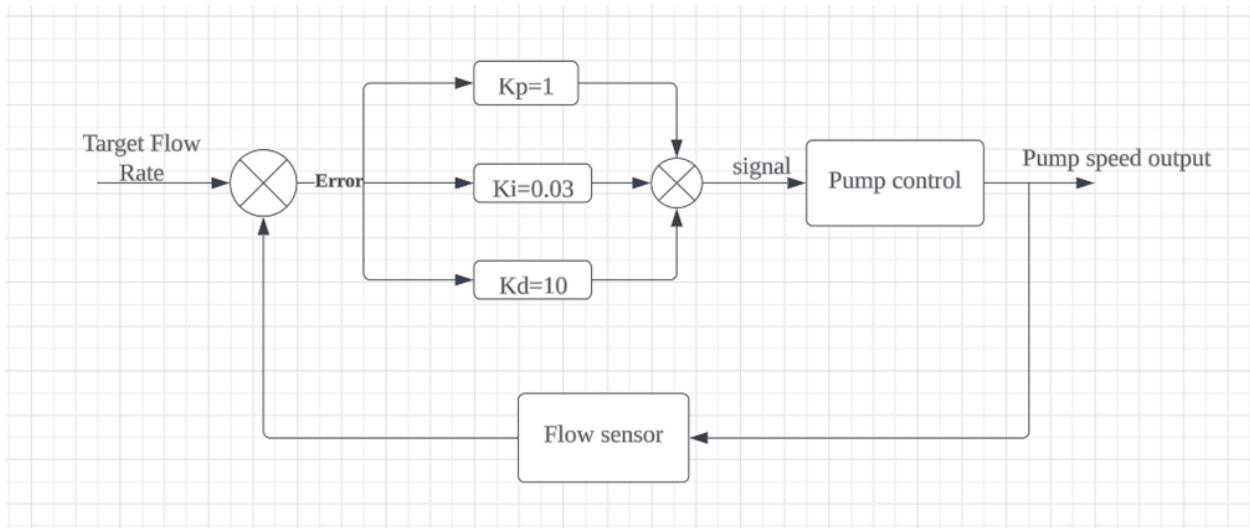


Figure 6.5.1.2.1

The PID control system we designed is:

$$P = K_P + \frac{K_I}{s} + K_D s \quad (K_P = 1, K_I = 0.01, K_D = 10) \quad (6.5.1.2.1)$$

We drew the step response diagram before and after flow control. Through the feedback system of the PID controller and sensor, the overshoot problem of flow was solved. We continue with the same duty cycle to make the pump work, record the data and plot the step response:

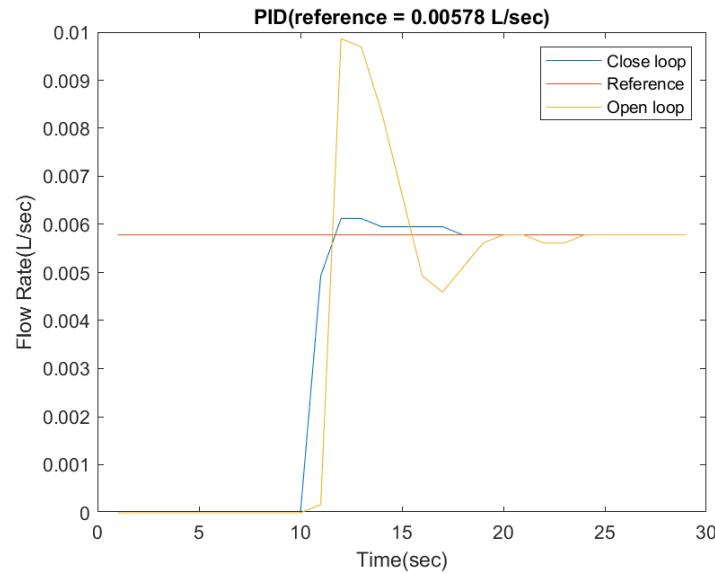


Figure 6.5.1.2.2 After PID control

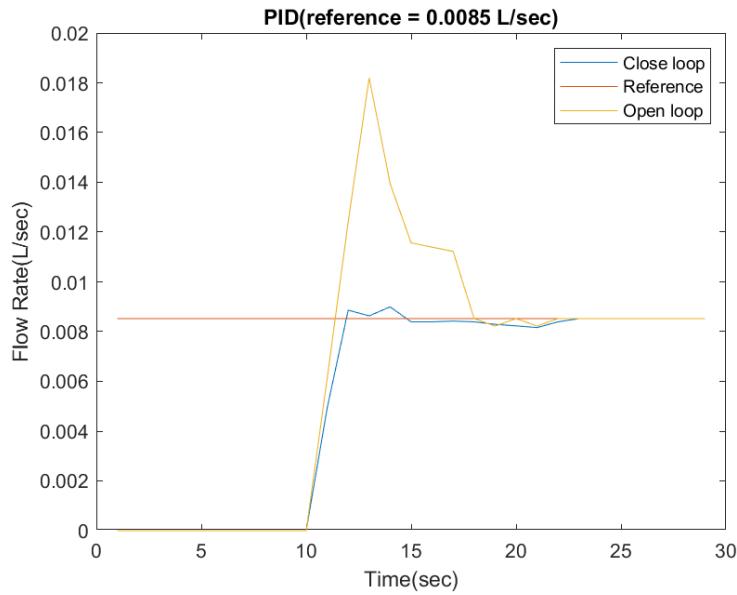


Figure 6.5.1.2.3 After PID control

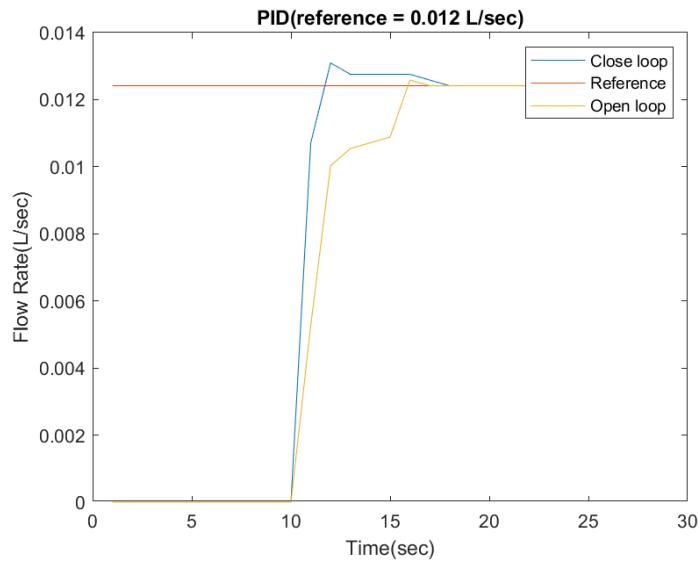


Figure 6.5.1.2.4 After PID control

After optimization of the PID controller and feedback system, the flow rate of the water pump can be stabilized.

6.5.2 Camera

Lesion Identification Model VS Crop Identification Model

The following two pictures are demonstration pictures of the lesion identification model and crop identification model respectively.



Figure 6.5.2.1.1 Lesion Identification Model



Figure 6.5.2.1.2 Crop Identification Model

Figure 6.5.2.1.1 (lesion identification model) only identifies leaf lesions but not crops. *Figure 6.5.2.1.2* (crop identification model) only identifies crops. These two models are suitable for different situations. Users can choose to treat only diseased leaves to prevent the spread of disease. Or choose to identify crops to spray pesticide and prevent potential leaf lesions.

Image recognition model accuracy test

For several crops we planted, 2 specific image recognition models (based on YOLOv5s) were trained. One of them is used to identify whether there are leaf lesions on the crop, while the other is

used to identify whether there are surrounding crops. Although compared to the previous model, it supports the identification of multiple crops and multiple leaf lesions. The accuracy of these newly trained models was significantly higher (One of the new models (lesion identification) mAP@50 is 0.995 and the other is 0.995(crop identification), the previous model only has 0.628), and when they are applied in actual applications, their performance is better.

Model-1 (Lesion Identification) accuracy test

We simulated leaf lesions by painting spots on the leaves. The following picture is a demonstration of the model automatically marking leaf lesions with red borders after identifying the test sample set.



Figure 6.5.2.2.1.1 Test Demonstration

We tested a total of 64 test samples with diseased leaves. The model accurately predicted the diseased leaves in these 64 samples. The precision of the model reached 99.4% *Figure 6.5.2.2.1.2*.

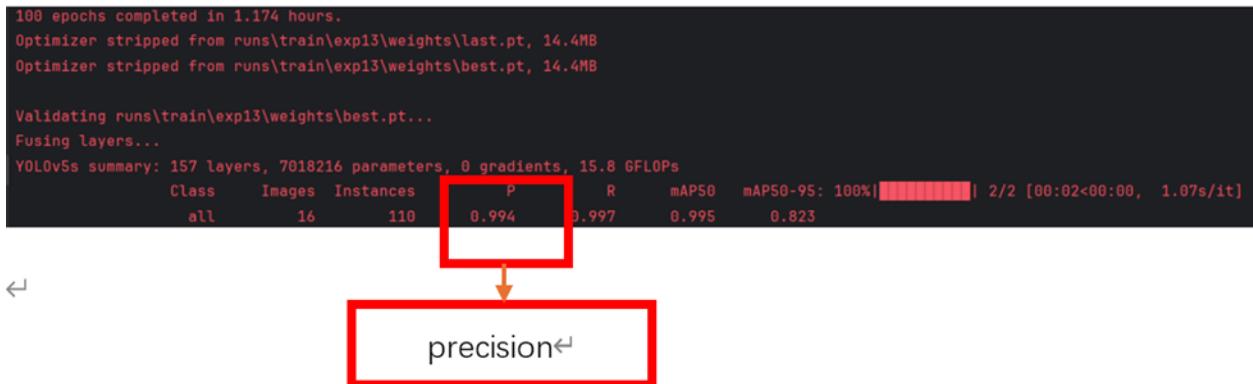


Figure 6.5.2.2.1.2 Precision of The Model

Model-2 (Crop Identification) accuracy test

We verified the accuracy of this model on our potted crops. The figure below is a demonstration of the model automatically marking crops with a red border after identifying the test sample set.



Figure 6.5.2.2.2.1 Crop Identification Demo

We tested a total of 79 test samples, and the model accurately predicted the crops in these 79 samples. The precision of the model reached 99.8% *Figure 6.5.2.2.1.4*.



```

100 epochs completed in 1.497 hours.
Optimizer stripped from runs\train\exp15\weights\last.pt, 14.4MB
Optimizer stripped from runs\train\exp15\weights\best.pt, 14.4MB

Validating runs\train\exp15\weights\best.pt...
Fusing layers...
YOLOv5s summary: 157 layers, 7012822 parameters, 0 gradients, 15.8 GFLOPs
      Class    Images Instances      P      R   mAP50   mAP50-95: 100% | 1/1 [00:01<00:00,  1.66s/it]
      all         9       36    0.998      1    0.995    0.615
  
```

precision ↵

Figure 6.5.2.2.1 Precision Of Model

Area Measurement Accuracy Test

We first use the lesion identification model to measure the area of the diseased leaves, as shown in the figure below.

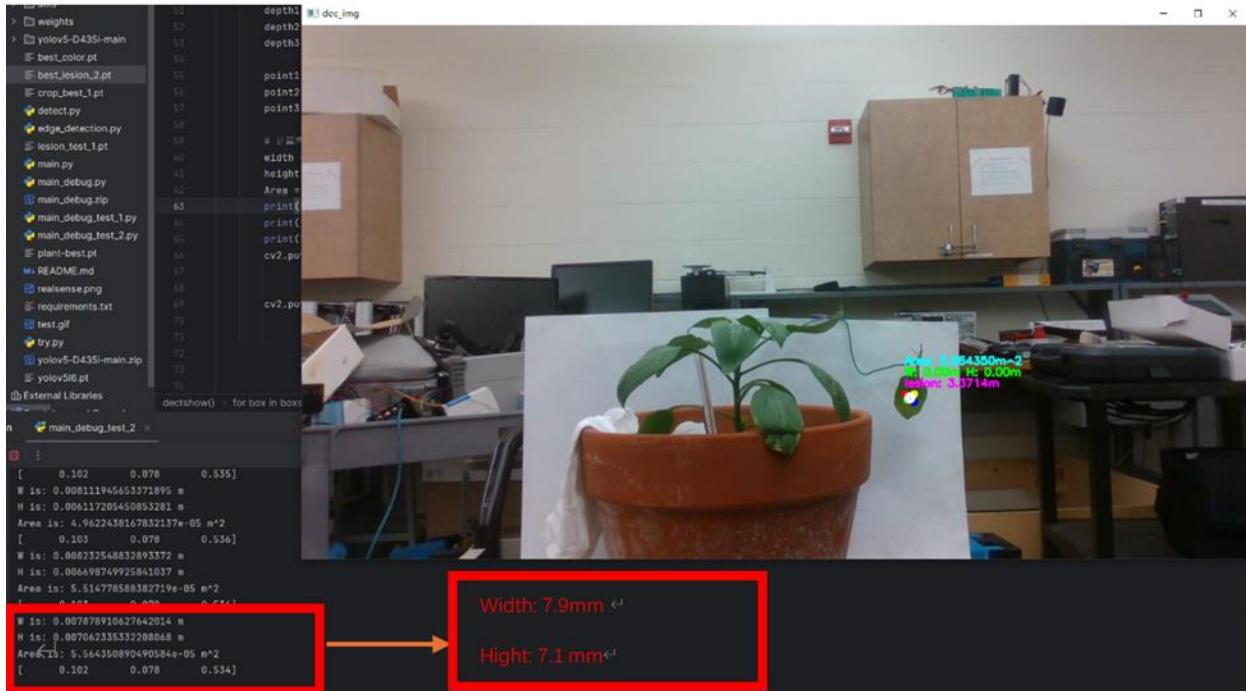


Figure 6.5.2.2.3.1 Diseased Leaves Measurement

Based on the picture above, it can be noticed that in the lower left corner of the picture (inside the red rectangular mark box), the data printed out on the terminal is the width(7.9mm) and

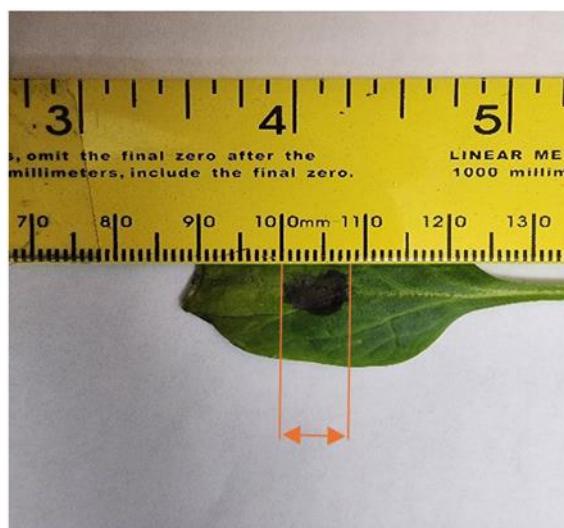
height(7.1mm) of the target measured by the model, as well as the X(0.102m),Y(0.078m),Z(0.534m) target position. Since the area of the diseased leaves is really small, the calculated data are very small decimals. As a result, the data marked on the display interface cannot display all the numbers, which makes it looks strange.

The picture below shows the actual measured width and height of leaf lesions.



← lesion leaf width(8mm)

Figure 6.5.2.2.3.3 Actual Measured Height Of Lesion
Figure 6.5.2.2.3.2 Actual Measured width Of Lesion



← lesion leaf height(8mm)

Based on the above data from pictures, the area of the diseased leaves can be roughly calculated (about 64 mm^2). And the area calculated by the model is approximately 56.09 mm^2 . The error is about 12.35%. Considering that the coverage area of the nozzle at this distance (about 0.25m far away from target after considering the movement of robotic arm) is about 9637 mm^2 (for calculation details, please refer to the nozzle part), which is enough to cover the impact of the error.

We then use the crop identification model to measure the area of the crop, as shown in the figure below.



Figure 6.5.2.2.3.4 Crop Identification Model Demo

Based on the picture above *Figure 6.5.2.2.3.4*, it shows width (18 cm) and height (10 cm) of the target measured by the model.



Figure 6.5.2.2.3.5 Actual Measured Width



Figure 6.5.2.2.3.6 Actual Measured Height

Based on the above data from pictures (*Figure 6.5.2.2.3.5* and *Figure 6.5.2.2.3.6*), the area of the diseased leaves can be roughly calculated (about 156.24 cm^2). And the area calculated by the model is approximately 197.51 cm^2 . The error is about 26.45%. Considering that the coverage area of the nozzle at this distance (about 0.6851m far away from target after considering the movement of robotic arm) is about 723.56 cm^2 (for calculation details, please refer to the nozzle part), which is enough to cover the impact of the error.

Position Measurement Accuracy Test

Considering the working space of the robotic arm, the testing range is limited from 30 cm to 60 cm (camera ideal working range: 30 cm - 300 cm from datasheet). Every 5cm, the distance measured by camera are compared with the actual distances. *Figure 6.5.2.2.4.1* shows how the test is conducted.



Figure 6.5.2.2.4.1 Position Measurement Accuracy Test

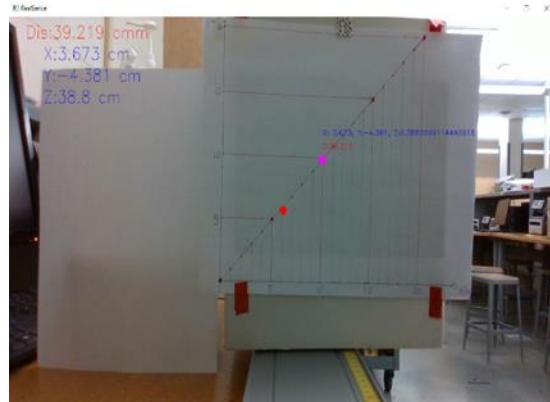


Figure 6.5.2.2.4.2 Position Measurement Accuracy Test

In *Figure 6.5.2.2.4.2*, this rectangular border represents the resolution of the current camera, and the red dot in the middle represents the coordinate center point of the camera. A new coordinate system is recreated to represent the actual coordinates in the actual world (coordinates on white paper in background). Now on the image, the camera origin is at about (6, 6) cm. Selecting a point at (10, 10) cm, and then read the value detected by the camera. The actual value should be (4, 4, 40) cm, the z-axis distance is obtained from the guide rail with labeled distance, and the position camera measured

Table 6.5.2.2.4.1 Errors At Different Distances

is $(3.673, -4.381, 38.8)$ cm, the reason for the negative sign is that the y-axis of the camera's coordinate axis is opposite. The following table shows the errors detected at different distances.

Distance(cm)	30	35	40	45	50	55
Errors	1.87%	1.70%	1.15%	1.24%	1.50%	1.37%

The measured errors by camera are less than 2%, which can provide good data to robotic arm to reach the target. And now the camera supported resolution is up to 1280×720 pixels, which meets the objective 640×480 pixels.

6.5.3 Robotic arm

Coordinate Conversion Test

The first is the test of coordinate conversion. We mark the coordinates of the robot arm and the coordinates of the camera (shown in *Figure 6.5.3.1.1*), and then get the coordinate conversion formula:

$$T = Trans_{x,d} \cdot Trans_{z,h} \cdot Rot_{z,-90^\circ} \cdot Rot_{x,-90^\circ} \quad (6.5.3.1.1)$$

$$T = \begin{bmatrix} 1 & 0 & 0 & d \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & h \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos(-90^\circ) & -\sin(-90^\circ) & 0 & 0 \\ \sin(-90^\circ) & \cos(-90^\circ) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(-90^\circ) & -\sin(-90^\circ) & 0 \\ 0 & \sin(-90^\circ) & \cos(-90^\circ) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.5.3.1.2)$$

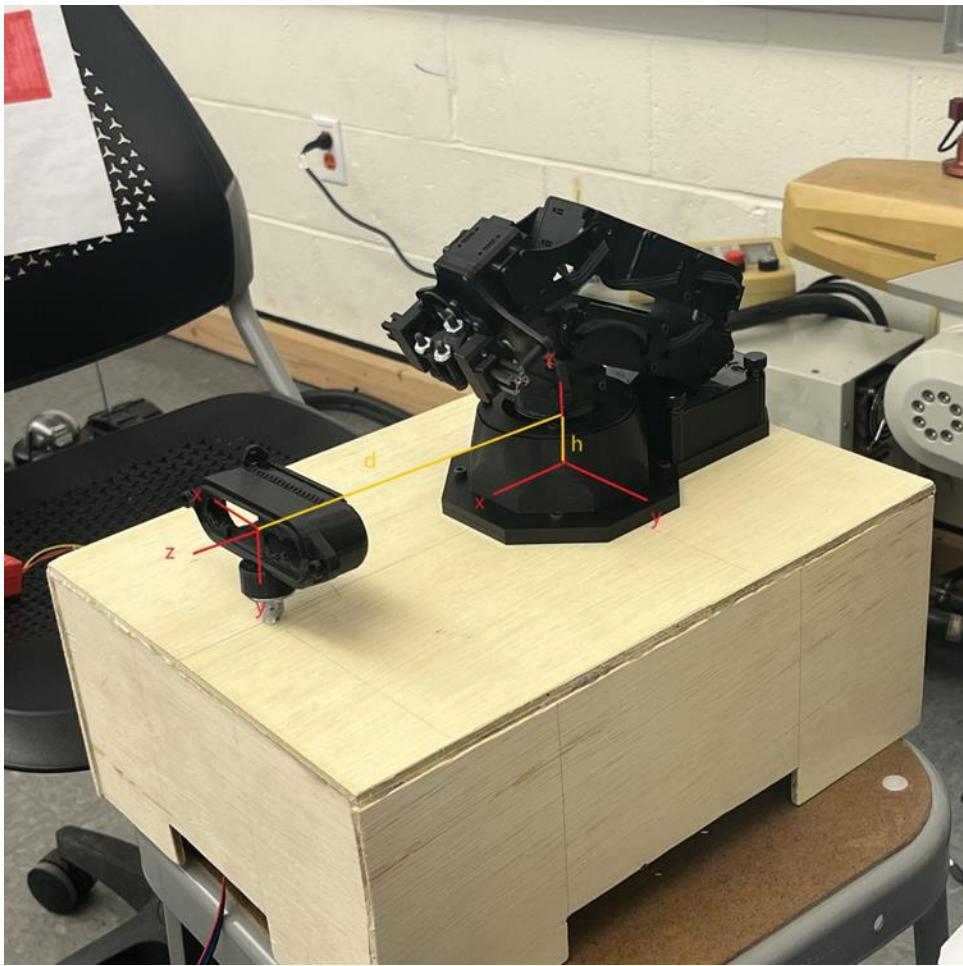


Figure 6.5.3.1.1 Coordinate Conversion

Then the target position is obtained through the image recognition model, then converted into the target position based on the robot arm coordinates through the code, and compared with the target position based on the robot arm coordinates that we measured, we can get the following *Table 6.5.3.1.1.*

Table 6.5.3.1.1 Error Between Target Position And Converted Position

	Target positions based on camera	Target positions based on robotic arm (converted)	Actual target positions based on the robot arm	Error
90°	[-0.099 -0.096 0.68]	0.081 0.68 0.136	0.055 0.676 0.133	[0.0260' 0.0040' 0.0030']
	[-0.099 -0.096 0.68]	0.081 0.68 0.136	0.055 0.676 0.133	[0.0260' 0.0040' 0.0030']
	[-0.099 -0.097 0.683]	0.081 0.683 0.137	0.054 0.677 0.133	[0.0270' 0.0060' 0.0040']
	[-0.099 -0.097 0.683]	0.081 0.683 0.137	0.054 0.677 0.133	[0.0270' 0.0060' 0.0040']
	[-0.101 -0.098 0.691]	0.079 0.691 0.138	0.054 0.68 0.133	[0.0250' 0.0110' 0.0050']
	[-0.101 -0.098 0.691]	0.079 0.691 0.138	0.054 0.68 0.133	[0.0250' 0.0110' 0.0050']
	[-0.101 -0.097 0.689]	0.079 0.689 0.137	0.055 0.671 0.132	[0.0240' 0.0180' 0.0050']
	[-0.101 -0.097 0.689]	0.079 0.689 0.137	0.055 0.671 0.132	[0.0240' 0.0180' 0.0050']
	[-0.101 -0.098 0.689]	0.079 0.689 0.138	0.054 0.679 0.133	[0.0250' 0.0100' 0.0050']
	[-0.101 -0.098 0.689]	0.079 0.689 0.138	0.054 0.679 0.133	[0.0250' 0.0100' 0.0050']
	[-0.1 -0.098 0.685]	0.08 0.685 0.138	0.054 0.678 0.133	[0.0260' 0.0070' 0.0050']
	[-0.1 -0.098 0.685]	0.08 0.685 0.138	0.054 0.678 0.133	[0.0260' 0.0070' 0.0050']
	[-0.101 -0.099 0.691]	0.079 0.691 0.139	0.055 0.674 0.132	[0.0240' 0.0170' 0.0070']
	[-0.101 -0.099 0.691]	0.079 0.691 0.139	0.055 0.674 0.132	[0.0240' 0.0170' 0.0070']
	[-0.1 -0.098 0.687]	0.08 0.687 0.138	0.055 0.674 0.132	[0.0250' 0.0130' 0.0060']
	[-0.1 -0.098 0.687]	0.08 0.687 0.138	0.055 0.674 0.132	[0.0250' 0.0130' 0.0060']
	[-0.1 -0.097 0.688]	0.08 0.688 0.137	0.054 0.678 0.133	[0.0260' 0.0100' 0.0040']
	[-0.101 -0.098 0.69]	0.08 0.688 0.137	0.054 0.678 0.133	[0.0260' 0.0100' 0.0040']
	[-0.101 -0.098 0.69]	0.08 0.688 0.138	0.054 0.681 0.133	[0.0250' 0.0090' 0.0050']
	[-0.1 -0.097 0.688]	0.079 0.689 0.138	0.054 0.681 0.133	[0.0250' 0.0090' 0.0050']
	[-0.1 -0.097 0.688]	0.08 0.688 0.137	0.054 0.68 0.133	[0.0260' 0.0080' 0.0040']
	[-0.099 -0.097 0.685]	0.08 0.688 0.137	0.054 0.68 0.133	[0.0260' 0.0080' 0.0040']
	[-0.099 -0.097 0.685]	0.081 0.685 0.137	0.054 0.679 0.133	[0.0270' 0.0060' 0.0040']
	[-0.1 -0.097 0.685]	0.081 0.685 0.137	0.054 0.679 0.133	[0.0270' 0.0060' 0.0040']
	[-0.1 -0.097 0.685]	0.08 0.685 0.137	0.055 0.676 0.133	[0.0250' 0.0090' 0.0040']
	[-0.1 -0.097 0.686]	0.08 0.685 0.137	0.055 0.676 0.133	[0.0250' 0.0090' 0.0040']
	[-0.1 -0.097 0.686]	0.08 0.686 0.137	0.054 0.679 0.133	[0.0260' 0.0070' 0.0040']
	[-0.101 -0.097 0.689]	0.08 0.686 0.137	0.054 0.679 0.133	[0.0260' 0.0070' 0.0040']
	[-0.101 -0.097 0.689]	0.079 0.689 0.137	0.053 0.683 0.134	[0.0260' 0.0060' 0.0030']
	[-0.101 -0.098 0.691]	0.079 0.689 0.137	0.053 0.683 0.134	[0.0260' 0.0060' 0.0030']
	[-0.101 -0.098 0.691]	0.079 0.691 0.138	0.055 0.676 0.132	[0.0240' 0.0150' 0.0060']
	[-0.101 -0.098 0.69]	0.079 0.691 0.138	0.055 0.676 0.132	[0.0240' 0.0150' 0.0060']
-90°	[-0.101 -0.098 0.69]	0.281 -0.69 0.138	0.306 -0.68 0.133	[-0.0250' -0.0100' 0.0050']
	[-0.101 -0.098 0.691]	0.281 -0.69 0.138	0.306 -0.68 0.133	[-0.0250' -0.0100' 0.0050']
	[-0.101 -0.098 0.691]	0.281 -0.691 0.138	0.306 -0.681 0.133	[-0.0250' -0.0100' 0.0050']
	[-0.1 -0.097 0.684]	0.281 -0.691 0.138	0.306 -0.681 0.133	[-0.0250' -0.0100' 0.0050']
	[-0.1 -0.097 0.684]	0.28 -0.684 0.137	0.306 -0.678 0.133	[-0.0260' -0.0060' 0.0040']
	[-0.1 -0.097 0.687]	0.28 -0.684 0.137	0.306 -0.678 0.133	[-0.0260' -0.0060' 0.0040']
	[-0.1 -0.097 0.687]	0.28 -0.687 0.137	0.306 -0.679 0.133	[-0.0260' -0.0080' 0.0040']
	[-0.1 -0.097 0.691]	0.28 -0.687 0.137	0.306 -0.679 0.133	[-0.0260' -0.0080' 0.0040']
	[-0.1 -0.099 0.691]	0.28 -0.691 0.139	0.306 -0.679 0.133	[-0.0260' -0.0120' 0.0060']
	[-0.1 -0.098 0.692]	0.28 -0.691 0.139	0.306 -0.679 0.133	[-0.0260' -0.0120' 0.0060']
	[-0.1 -0.098 0.692]	0.28 -0.692 0.138	0.306 -0.673 0.132	[-0.0260' -0.0190' 0.0060']
	[-0.101 -0.098 0.69]	0.28 -0.692 0.138	0.306 -0.673 0.132	[-0.0260' -0.0190' 0.0060']
	[-0.101 -0.098 0.69]	0.281 -0.69 0.138	0.306 -0.679 0.133	[-0.0250' -0.0110' 0.0050']
	[-0.101 -0.098 0.69]	0.281 -0.69 0.138	0.306 -0.679 0.133	[-0.0250' -0.0110' 0.0050']
	[-0.101 -0.098 0.69]	0.281 -0.69 0.138	0.306 -0.679 0.133	[-0.0250' -0.0110' 0.0050']
	[-0.1 -0.097 0.688]	0.281 -0.69 0.138	0.305 -0.676 0.133	[-0.0240' -0.0140' 0.0050']
	[-0.1 -0.097 0.688]	0.28 -0.688 0.137	0.305 -0.674 0.132	[-0.0250' -0.0140' 0.0050']
	[-0.101 -0.098 0.692]	0.28 -0.688 0.137	0.305 -0.674 0.132	[-0.0250' -0.0140' 0.0050']
	[-0.101 -0.098 0.692]	0.281 -0.692 0.138	0.306 -0.678 0.132	[-0.0250' -0.0140' 0.0060']
	[-0.099 -0.097 0.688]	0.281 -0.692 0.138	0.306 -0.678 0.132	[-0.0250' -0.0140' 0.0060']
	[-0.099 -0.097 0.688]	0.279 -0.688 0.137	0.306 -0.678 0.133	[-0.0270' -0.0100' 0.0040']
	[-0.101 -0.098 0.689]	0.279 -0.688 0.137	0.306 -0.678 0.133	[-0.0270' -0.0100' 0.0040']
	[-0.101 -0.098 0.689]	0.281 -0.689 0.138	0.306 -0.678 0.133	[-0.0250' -0.0110' 0.0050']
	[-0.101 -0.098 0.693]	0.281 -0.689 0.138	0.306 -0.678 0.133	[-0.0250' -0.0110' 0.0050']
	[-0.101 -0.098 0.693]	0.281 -0.693 0.138	0.306 -0.679 0.133	[-0.0250' -0.0140' 0.0050']
	[-0.099 -0.097 0.683]	0.281 -0.693 0.138	0.306 -0.679 0.133	[-0.0270' -0.0040' 0.0050']
	[-0.099 -0.097 0.683]	0.279 -0.683 0.137	0.306 -0.679 0.132	[-0.0270' -0.0040' 0.0050']
	[-0.1 -0.098 0.688]	0.279 -0.683 0.137	0.306 -0.679 0.132	[-0.0270' -0.0040' 0.0050']
	[-0.1 -0.098 0.688]	0.28 -0.688 0.138	0.307 -0.682 0.134	[-0.0270' -0.0060' 0.0040']
	[-0.101 -0.097 0.689]	0.28 -0.688 0.138	0.307 -0.682 0.134	[-0.0270' -0.0060' 0.0040']
	[-0.101 -0.097 0.689]	0.281 -0.689 0.137	0.307 -0.683 0.134	[-0.0260' -0.0060' 0.0030']
	[-0.101 -0.097 0.689]	0.281 -0.689 0.137	0.307 -0.683 0.134	[-0.0260' -0.0060' 0.0030']

From the table, we can see that there is a certain error between the actual target position and the converted target position. The maximum error in the x direction is between 2.4cm (about 0.94 in) and 2.7cm (about 1.06 in), and the error in the y direction is between 0.04cm (about 0.02 in) and 1.9cm (about 0.75 in). The error in the z direction is the smallest between 0.03cm (about 0.01 in) and 0.07cm (about 0.03 in). The reason for the larger errors in the x-direction and y-direction is that the values of

the horizontal distance 'd' between the robotic arm and the camera, and the vertical height difference 'h', were measured manually, which introduces some degree of error.

Transmission Coordinate Test

Then, we also conducted camera transmission tests, as shown in the figure:

```

YOLOV5-D435I-MAIN
├── __pycache__
├── data
├── models
├── runs
├── utils
└── weights
    ├── a-b.py
    └── a.txt
    └── b.txt
    └── best_color.pt
    └── best.pt
    └── detect.py
    └── lesion_test_1.pt
    └── main_debug_1.py
    └── main_debug_2.py
    └── main_debug_3.py
    └── main_debug_4.py
    └── main_debug_5.py
    └── main_debug_6.py
    └── main_debug_7.py
    └── main_debug_7X.py
    └── main_debug_Barea.py
    └── main_debug.py
    └── main.py
    └── matrixtest.py
    └── output.txt
    └── plant-best.pt
    └── position_control_px100.py
    └── README.md
    └── realsense.png
    └── requirements.txt
    └── script.py
    └── test.gif
    └── test.py
    └── try.py
    └── yolov5l6.pt

```

Figure 6.5.3.2.1 Coordinates From Image Recognition

```

DESIGN
├── control_v1.py
├── control_v2.py
├── control_v3.py
├── control_v4.py
├── control_v5.py
├── control_v6.1.py
├── control_v6.2_test.py
├── control_v6.2.py
├── data.txt
├── joint_control_px100.py
└── output.txt

```

	0.055, 0.676, 0.133
1	0.055, 0.676, 0.133
2	0.055, 0.676, 0.133
3	0.054, 0.677, 0.133
4	0.054, 0.677, 0.133
5	0.054, 0.68, 0.133
6	0.054, 0.68, 0.133
7	0.055, 0.671, 0.132
8	0.055, 0.671, 0.132
9	0.054, 0.679, 0.133
10	0.054, 0.679, 0.133
11	0.054, 0.678, 0.133
12	0.054, 0.678, 0.133
13	0.055, 0.674, 0.132
14	0.055, 0.674, 0.132
15	0.055, 0.674, 0.132
16	0.055, 0.674, 0.132
17	0.054, 0.678, 0.133
18	0.054, 0.678, 0.133
19	0.054, 0.681, 0.133
20	0.054, 0.681, 0.133
21	0.054, 0.68, 0.133
22	0.054, 0.68, 0.133
23	0.054, 0.679, 0.133
24	0.054, 0.679, 0.133
25	0.055, 0.676, 0.133
26	0.055, 0.676, 0.133
27	0.054, 0.679, 0.133
28	0.054, 0.679, 0.133
29	0.053, 0.683, 0.134
30	0.053, 0.683, 0.134
31	0.055, 0.676, 0.132
32	0.055, 0.676, 0.132
33	0.306, -0.68, 0.133
34	0.306, -0.68, 0.133
35	0.306, -0.681, 0.133
36	0.306, -0.681, 0.133
37	0.306, -0.678, 0.133
38	0.306, -0.678, 0.133
39	0.306, -0.679, 0.133
40	0.306, -0.679, 0.133
41	0.306, -0.679, 0.133
42	0.306, -0.679, 0.133
43	0.306, -0.673, 0.132
44	0.306, -0.673, 0.132
45	0.306, -0.679, 0.133
46	0.306, -0.679, 0.133
47	0.305, -0.676, 0.133
48	0.305, -0.676, 0.133
49	0.305, -0.674, 0.132
50	0.305, -0.674, 0.132
51	0.306, -0.678, 0.132
52	0.306, -0.678, 0.132
53	0.306, -0.678, 0.133
54	0.306, -0.678, 0.133
55	0.306, -0.678, 0.133
56	0.306, -0.678, 0.133
57	0.306, -0.679, 0.133
58	0.306, -0.679, 0.133
59	0.306, -0.679, 0.132
60	0.306, -0.679, 0.132

Figure 6.5.3.2.2 Coordinates Received By The Robotic Arm

Figure 6.5.3.2.2 displays the coordinates stored after camera measurement, while the Figure

6.5.3.2.1 shows the coordinates transmitted to the robotic arm code. It can be observed that there is no data loss or error.

One Target Test and Multiple Targets Test

Finally, put single-target tests and multiple targets tests together. I control the camera to execute 32 cycles at 90 degrees and -90 degrees respectively, that is, to obtain 32 sets of data, and then take the most stable set of data among the 32 sets of data as the input of the robotic arm. Then the area of the target is obtained, and the spraying distance is obtained through the formula of theoretical coverage, thereby obtaining the final spraying position. In the test of a single target, whether it is 90 degrees or -90 degrees, we can see that the error between the actual position reached by the end of the robotic arm and the calculated spray position is within 0.7cm (about 0.28 in). In the test of multiple targets, the position error was also within 0.5cm (about 0.2 in). These errors are within our acceptable range, even smaller than our expected error of 1cm (shown in *Table 6.5.3.3.1*).

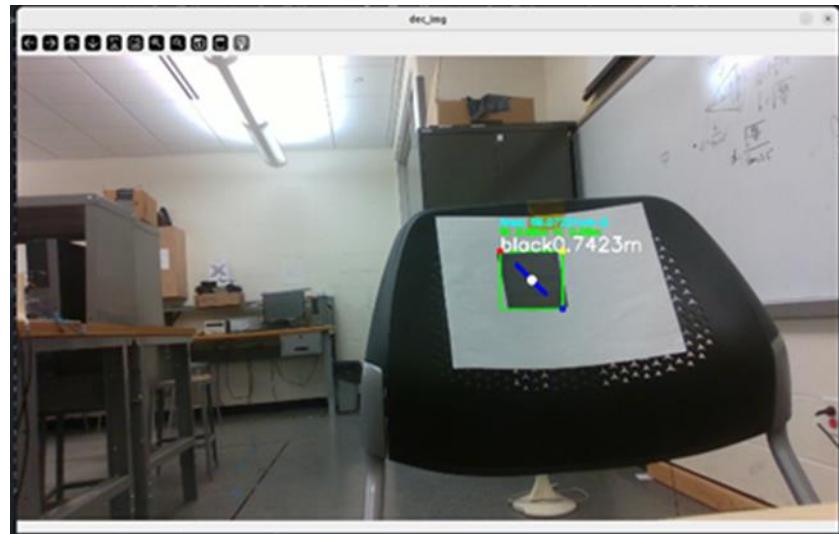


Figure 6.5.3.3.1 Single Target Test

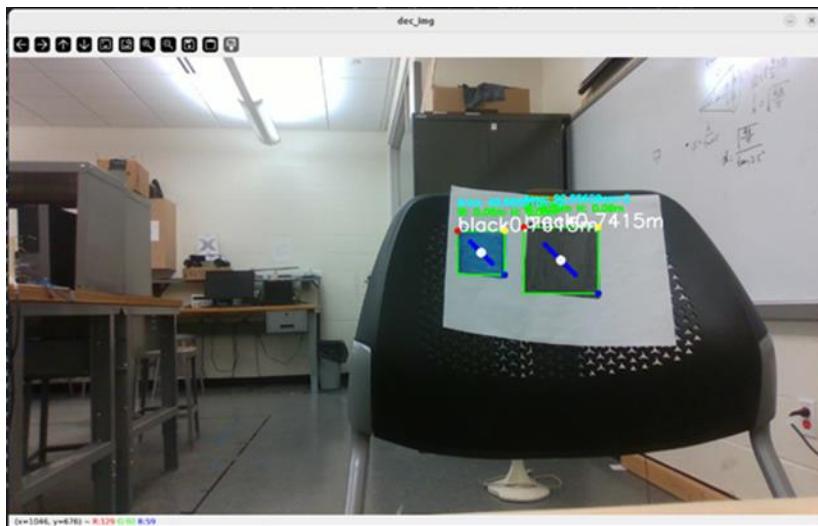


Figure 6.5.3.3.2 Multiple Target Test

Table 6.5.3.3.1 Error Between the Actual Position Reached By The End Effector And The Calculated Spray Position

one target	the target position	
	90° X(m), Y(m), Z(m), Area(cm^2)	-90° X(m), Y(m), Z(m), Area(cm^2)
Distance(m)	0.421638871	0.421638871
Spray Position(m)	0.11311493460474364, 0.27724248677633245, 0.0184828324517555	0.021946748636258565, - 0.29847578145311643, 0.020749653256099003
Actual Position(m)	0.12, 0.28, 0.02	0.02, -0.3, 0.02
Error(m)	0.007, 0.003, 0.002	0.001, 0.002, 0.001

two targets	the target position	
	90° X(m), Y(m), Z(m), Area(cm^2)	-90° X(m), Y(m), Z(m), Area(cm^2)
The first one target		
Distance(m)	0.329387495	0.329387495
Spray Position(m)	0.1299719124501072, 0.2688704672388207, 0.028565255483540046	-0.0011867961037220843, - 0.29828142073548386, 0.03204349480049628]

Actual Position(m)	0.13, 0.27, 0.03	0, -0.3, 0.03
Error(m)	0, 0.002, 0.001	0.001, 0.002, 0.002
The second one target		
Distance(m)	0.329387495	0.329387495
Spray Position(m)	0.09780233437790065, 0.2817009322236057, 0.03285252158639906	0.03970712907542338, - 0.2952480587687421, 0.03538259026522875
Actual Position(m)	0.1, 0.28, 0.03	0.04, -0.3, 0.03
Error(m)	0.003, 0.002, 0.002	0, 0.005, 0.005

Replacement of Improved Recognition Model Test

The upgraded recognition model can identify the real area and coordinates of crops. We use this improved model for testing, which is divided into two types: single-target tests and multiple-target tests. Based on the findings from previous tests, where the error results for rotating the camera by 90 degrees and -90 degrees were quite similar, we decided to omit the -90 degrees rotation test and only conduct the 90 degrees rotation test. Initially, we test a single target by capturing data from 10 sets of average performance using the camera, as shown in the following *Table 6.5.3.4.1*:

Table 6.5.3.4.1 Error in one target test

One target	The target positions
	X(m), Y(m), Z(m), Area(cm ²)
Distance(m)	0.148445265
Spray Position(m)	0.10169521996310545, 0.28112605309172606, 0.025024478210816523
Actual Position(m)	0.1, 0.28, 0.02
Error(m)	0.001, 0.001, 0.005

From the table above, we can see that the errors on the x, y, z axis are within 0.5cm (about 0.2 in), which is within our error range.

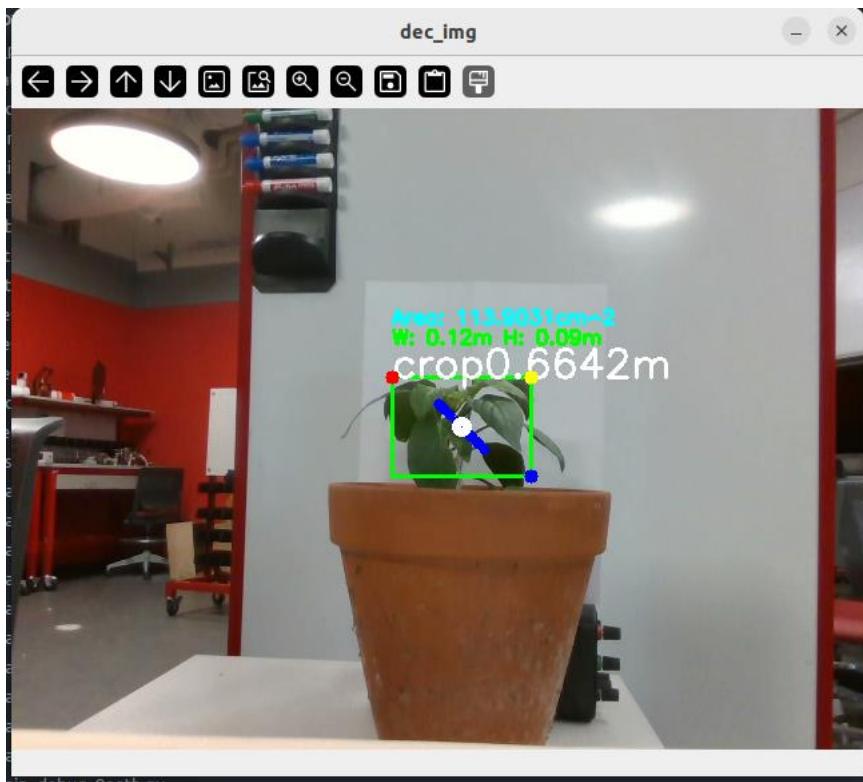


Figure 6.5.3.4.1 Single Target Test

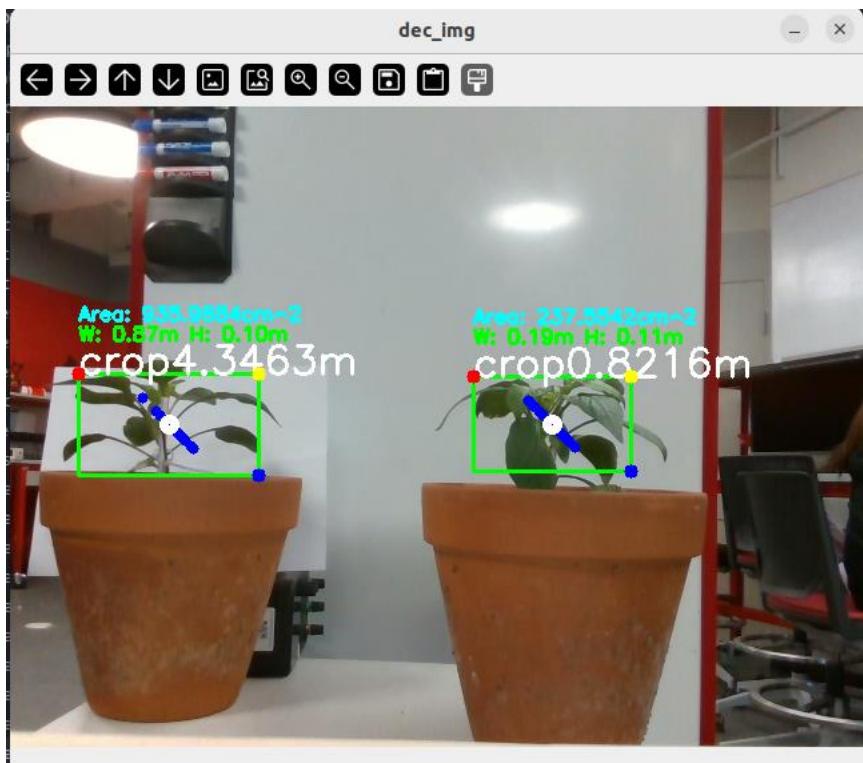


Figure 6.5.3.4.2 Multiple Target Test

Then we conduct multiple target tests. Here we select two crops and obtain twelve groups of average performance data, as shown in the following *Table 6.5.3.4.2*:

Table 6.5.3.4.2 Error in multiple targets test

The first one target	
Distance(m)	0.214939294
Spray Position(m)	0.12634366485694695, 0.27109172183872476, 0.023378552158568922
Actual Position(m)	0.13, 0.27, 0.02
Error(m)	0.004, 0.001, 0.003
The second one target	
Distance(m)	0.214939294
Spray Position(m)	-0.02261817532403405, 0.2983859283132185, 0.021313280593801318
Actual Position(m)	-0.2, 0.3, 0.02
Error(m)	0.002, 0.002, 0.001

We can see that the error of one crop is within 0.4cm (about 0.16 in) and the error of the other crop is within 0.2cm (about 0.08 in). This is very similar to our single crop test results and is also within our expected error range.

7 Final Product Design

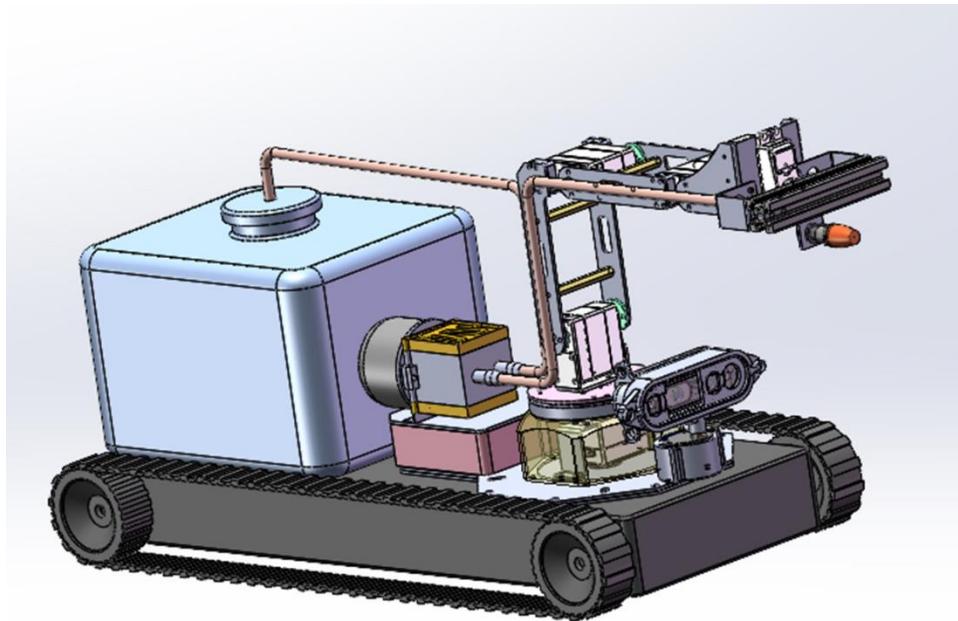


Figure 7.1 Final design



Figure 7.2 Final product

After testing the Alpha prototype, we made several improvements to the robotic arm, focusing on refining the code that controls its movements to enhance precision. For the battery section, we replaced the two batteries with a 24,000mAh portable power bank, enabling the product to support prolonged continuous operation. Regarding the housing, we used a wooden box to conceal the complex internal wiring, thus preventing clutter and damage to the circuits.

Following the testing of the Beta prototype, we did not make any substantial improvements. This could indicate that the enhancements implemented post-Alpha testing were effective, or it might suggest that further evaluations are needed to identify areas requiring significant changes.

8 Miscellaneous

8.1 Cost Analysis

Table 8.1.1 Total cost

1. Components	Quantity	Book price	Total
Robot Arm (4DOF)	1	\$700.00	\$700.00
Nozzle	1	\$5.00	\$5.00
Pump	1	\$70.00	\$70.00
Depth Camera	1	\$300.00	\$300.00
Arduino Mega	1	\$30.00	\$30.00
Pesticide Reservoir	1	\$25.00	\$25.00
Stepper Motor	1	\$10.00	\$10.00
Battery	1	\$50.00	\$50.00
LCD	1	\$10.00	\$10.00
Flow Rate Sensor	1	\$10.00	\$10.00
Piping	1	\$10.00	\$10.00
Miscellaneous	1	\$10.00	\$10.00
Indirect costs (35% of all material)		35%	\$430.50
MATERIAL			\$1,230.00
2. LABOR	Set Up	Run	Cost/Hour
	(# of Min)	(# of Min)	
Assembly	15	30	\$30.00
Machining	10	25	\$35.00
Welding	10	20	\$32.00
Labeling	10	20	\$18.00
Labor			\$67.92
3. OVERHEAD			100%
(ON MATERIAL & LABOR COSTS)			\$1,297.92
			Subtotal
			\$2,595.83
		Profit	25%
			\$648.96
			4. Price
			\$3,244.79

In the development of our precision pesticide spraying robot, a detailed cost analysis has been conducted to ensure efficient budget allocation and to highlight the financial investment in key technological components. The robot arm, the most critical element for precise maneuvering, represents the highest expense at \$700. This is followed by the depth camera at \$300, essential for detecting crops and

calculating distances for effective spraying. Other vital components include the pump (\$70) which facilitates the pesticide delivery system, and the Arduino Mega (\$30) that serves as the control hub for integrating various functionalities. Additional components such as the battery, pesticide reservoir, stepper motor, LCD, flow rate sensor, and piping each cost between \$5 and \$50, cumulatively contributing to the system's operational efficiency. The total direct material costs amount to \$1,230, with indirect costs—accounting for additional assembly and operational contingencies—adding another 35%, or \$430.50, bringing the overall materials budget to \$1,660.50. This financial outline underscores our commitment to leveraging advanced technologies to enhance targeted pesticide application, while also managing resources responsibly within the projected budget.

8.2 Timeline



Figure 8.2.1 Timeline

This Gantt chart illustrates a commendable adherence to the set timeline, with all the planned tasks being completed ahead of schedule. The project initiated with the procurement of materials, followed by testing the components, which was finished by January 12th. Assembly of the mechanical structure was concluded by January 15th. A significant milestone was achieved on January 23rd with the integration of

electronics. The programming phase was completed promptly by February 2nd, and by February 4th, the team had successfully tested the basic functions. An analysis of the test results was conducted and completed by February 8th, and this led to the completion of the Milestone #1 report by February 9th. The subsequent phase focused on addressing any bugs, and this task was completed by February 14th.

The optimization and refinement of the project were finalized by March 8th, marking a pivotal phase of the project. On March 22nd, our team successfully completed the programming for image recognition and position measurement. The integration of three control programs was achieved by March 25th. Following this, reliability testing was conducted and completed by March 26th, and further analysis of the test results was finished by March 28th. This thorough testing phase culminated in the completion of the Milestone #2 report on March 29th.

In the early stages of April, the team undertook a series of field tests for the Precision Pesticide Spraying Robot. Despite our careful planning and prior successes, these initial tests did not go as well as expected, revealing several unforeseen issues. In response to these challenges, we promptly revisited our code and implemented the necessary modifications to address the identified problems. After refining our approach, we conducted a second round of field tests. The results of these subsequent tests were much more favorable, meeting our standards and expectations for the project.

Overall, the project not only adhered to the set deadlines but also expedited the completion of each phase, demonstrating the team's efficiency and dedication to the project's success.

8.3 Learning Plans and New Knowledges Acquired

Recognizing our limitations as non-experts in agriculture, we are committed to an exhaustive learning process that encompasses various domains crucial to the success of our project. Our approach to mastering the necessary skills and knowledge begins with engaging with experts in closely related fields. We plan to conduct interviews with crop science experts at Southern Illinois University Edwardsville (SIUE) to deepen our understanding of the agricultural needs and challenges that our robot aims to address. This interaction

will provide invaluable insights into crop behavior and the environmental factors affecting pesticide application.

Furthermore, we recognize the importance of the mechanical aspects of our design, particularly the fluid dynamics involved in pesticide spraying. To refine our spray system, we will consult a professor of fluid mechanics. This expertise will guide us in optimizing the spray mechanics to ensure efficient and effective pesticide distribution.

The technical development of our robotic system also requires high programming skills. We will undertake a comprehensive study of the ROS (Robot Operating System) tutorials to enhance our capabilities in programming and controlling the robotic arm. Additionally, mastering the programming of a depth camera is crucial for implementing the precise measurement functionalities our project demands.

Beyond hardware and software, we are also delving into 3D printing and modeling, essential for the custom design of robot components. Learning PID (Proportional-Integral-Derivative) control will further assist in refining the flow rate control precision. The integration of these components through the ROS2 package is another critical area of our study, ensuring that all subsystems work seamlessly together.

Our commitment to accumulating and integrating this diverse range of knowledge bases ensures that we can deliver a product that not only meets but exceeds our customers' expectations. The more resources we gather and learn from, the more confident we are in the reliability and effectiveness of our robotic solution, ultimately contributing to sustainable and precise agricultural practices.

8.4 Codes and Standards

Codes and Standards

ISO 22866:2012 - Agricultural and forestry machinery - Inspection of sprayers in use - Field

measurement of spray distribution: This standard provides guidelines for inspecting sprayers, which can be relevant for ensuring our design's spraying system distributes pesticides accurately.

ISO 16119 series - Agricultural and forestry machinery - Health and safety: These standards address health and safety aspects of agricultural machinery, ensuring that our product complies with necessary safety regulations.

ISO 13849 - Safety of machinery - Safety-related parts of control systems: This standard can be crucial for the safety of our robotic system, especially in terms of the control system.

EPA Regulations - United States Environmental Protection Agency: Adhere to local regulations regarding the use of pesticides. This includes regulations about the types of pesticides that can be used, application rates, and safety measures.

Incorporating Codes and Standards in Design Process

Safety Features: Design robot with safety features in accordance with ISO 13849 and other relevant safety standards. This may include emergency stop buttons, protective barriers, and fail-safe mechanisms.

Material Selection: Choose materials for the components (robotic arm, tubes, nozzle, etc.) that comply with any relevant material standards, especially those related to durability and chemical resistance.

Environmental Impact: Ensure that our design minimizes environmental impact, adhering to EPA regulations. This may involve using environmentally friendly pesticides and incorporating features to prevent overuse or accidental spills.

Precision and Accuracy: Align our design with ISO 22866 to ensure precision in pesticide application. This includes accurate positioning of the robotic arm and precise control of the spraying system.

Data Security and Privacy: If system involves data transmission and storage (e.g., crop health data), consider standards related to data security and privacy to protect sensitive information.

Incorporating Codes and Standards in Prototyping and Testing

Functional Testing: Test each component of our robot individually to ensure they function as intended. This includes testing the robotic arm's movement, camera's image recognition, and the spraying system's precision.

Safety Testing: Implement safety tests according to ISO 16119 and ISO 13849. This includes verifying emergency stop functionality, detecting and addressing potential hazards, and ensuring user safety.

Environmental Testing: Conduct tests to ensure our product can operate effectively in various environmental conditions, such as different types of crops, weather conditions, and terrains.

Calibration and Accuracy Testing: Calibrate the product regularly and perform accuracy testing to ensure that the robotic arm accurately targets leaves, and the spraying system delivers the correct amount of pesticides.

Compliance Documentation: Keep detailed documentation demonstrating how our design and prototype comply with each relevant code and standard. This will be valuable for regulatory approvals and quality assurance.

8.5 Ethical Issues and Societal Impacts

The main social impact of precision pesticide spraying robots is public health. The product aims to improve public health by reducing exposure to harmful chemicals through image recognition, microcontroller-based calculations and precise spraying. Precision spraying ensures that only the required amount of pesticide is used, minimizing the risk of pesticide residues on crops. This technology also allows for targeted treatment of specific areas, reducing the likelihood of pesticide drift, which can harm human health. Furthermore, the automation of spraying reduces the need for manual labor in pesticide application, protecting workers from potential health hazards associated with handling pesticides.

Safety and welfare are important to consumers. Consumers need to feel safe when using products and this will be ensured through extensive testing and research. This includes ensuring that all parts in contact with pesticides are corrosion-resistant to avoid leakage of pesticides, and to avoid electrical components coming into contact with liquids. Emergency stop systems and radar sensing devices should also be installed to avoid collisions with objects during operation. The product improves the welfare of agricultural workers and nearby communities by automating processes using robotic arms, reducing human exposure to pesticides, reducing the potential for pesticide drift and reducing the need for manual labor during pesticide application.

The design supports global and cultural factors by reducing the reliance on excessive pesticide use. Globally, it promotes sustainable and responsible agricultural practices, which align with global efforts to reduce the environmental impact of farming. Global impacts will include environmental factors such as pollution from production and transportation. Culturally, it respects the well-being of farming communities by reducing their exposure to harmful chemicals and minimizing the societal and cultural stigma associated with pesticide use.

The widespread adoption of precision pesticide spraying robots should have many positive social impacts. The main function of the product is to improve the efficiency, effectiveness, and sustainability of

pesticide application in agriculture. This product will not only increase the public health, safety and welfare. It will also enhance the productivity.

The traditional method of spraying pesticides is difficult to control the amount of pesticides used, which may lead to overuse of pesticides and harm the environment. The product can significantly benefit the environment by reducing pesticide usage. Precision spraying ensures that only the necessary amount of pesticide is applied, leading to less chemical runoff into water bodies and less soil contamination. This not only safeguards the local ecosystem but also contributes to long-term environmental sustainability by reducing the negative impacts of pesticide use on non-target organisms and biodiversity.

The technology has economic benefits by reducing the overall cost of pesticide application. Precision spraying reduces pesticide waste and saves on input costs. Additionally, it may improve crop yields and quality by ensuring that pests are efficiently managed, which can lead to increased agricultural productivity and economic prosperity for farmers. Furthermore, the reduced environmental impact can result in cost savings associated with environmental remediation and health care expenses, benefiting society as a whole.

Ethical and professional responsibilities are paramount for engineers. Engineers must consider the potential impact of their engineering solutions on various levels, including global, economic, environmental, and societal contexts. The foremost responsibility of an engineer is to prioritize the safety and well-being of the public. This means designing and implementing solutions that do not pose unnecessary risks to human life and the environment. Engineers should always follow best practices, codes, and standards to ensure safety. Engineers should consider the environmental impact of their projects and strive to design solutions that are environmentally sustainable. This includes minimizing waste, conserving resources, and mitigating the negative effects of projects on the environment. Engineers are expected to be honest and act with integrity in all their professional activities. This includes providing accurate information to clients and the public, disclosing any potential conflicts of interest, and not engaging in deceptive or fraudulent practices.

While not every engineering situation may require an equal emphasis on global, economic, environmental, and societal contexts, it is essential for engineers to be aware of these aspects in their decision-making process. Before any other consideration, we must first consider the economic impact that engineers design products can have. because economic factors play a fundamental role in the viability and success of any engineering project. Engineers need to assess whether the project can be completed within the allocated budget and whether it has the potential to generate returns on investment. Without a sound economic foundation, a project may not even get off the ground. While economic considerations are crucial, it's essential to note that they should not come at the expense of other important factors, such as safety, environmental impact, and societal well-being. A balanced approach that takes into account a wide range of considerations is the key to responsible and successful engineering.

9 Next Steps

In future work, we plan to use robotic arms with more degrees of freedom and larger sizes, such as the ViperX300S, to increase the workspace and perform more complex operations. For the camera part, we will use better-performing image training models such as YOLOv8 to improve model accuracy and recognition speed. At the same time, we also plan to train image recognition models that can recognize more types of crops to adapt to different tasks. For the nozzle part, we plan to use higher-precision water pumps and valves to ensure that the water pump can determine different amounts of pesticides according to the conditions of different plants.

Future improvements mainly involve integrating the existing system with a mobile robot base, which will allow for autonomous navigation throughout the farm. Adding navigation capabilities to the mobile base is crucial for enabling the robot to circumvent obstacles and effectively cover different areas without human intervention. Additionally, adopting an optimal controller for flow control will further enhance the precision of pesticide application, ensuring that each crop receives the precise dosage required without overuse or wastage, thereby maximizing efficacy and sustainability. Another significant upgrade involves transitioning from using traditional laptops for system control to more powerful, compact single-board

computers. This change will not only reduce the overall size and weight of the control system but also enhance its reliability and power efficiency, making the robot more practical and scalable for commercial use.

10 Conclusion

We conducted a preliminary analysis of our project in the first semester. We completed market research and background research on the application of pesticides in agricultural production. We decided to work on solving the problem of low precision and overuse of pesticides in the pesticide spraying process. We determined our final design goals based on our customer needs and suggestions. We built a block diagram to formulate the final workflow of the system, and a product architecture to build a 3D model of our ideal precision spraying robotic arm. We designed a variety of different product 3D models and compared them through Trade Study. We evaluated advantages and disadvantages between each model to get our final 3D design model. After setting the design goals and designing the 3D model, we split the system into three subsystems for technical analysis, which are the robotic arm, depth camera, and spray system. We found the most suitable solution based on customer needs and combined with theoretical design and actual conditions. Equipment for our system. Alternatives were developed for potential failures and failures, and a Gantt chart was developed to plan our project progress over the two semesters.

During the second semester, we began assembling and testing the first semester designs. First, we conducted FMEA analysis on potential system failures and failure factors. Then we divided the test into Alpha Prototype and Beta Prototype. In Alpha, we tested the accuracy of the moving position of the robotic arm, tested the deviation between the target position input by the robotic arm and the actual position, and simulated the spraying system and actual testing to find the optimal distance between the nozzle and the plant. In the Beta test, we conducted a coordinate conversion test between the camera and the robotic arm, a single coordinate conversion, and a multiple coordinate robotic arm movement test, and we also conducted a Replacement of Improved Recognition Model Test. For the camera, we completed the distance test, position test, surface area test, and image recognition model accuracy test. For the spray system, we

completed tests on flow control and accuracy testing on single spray volume. We completed all tests and connected all subsystems to realize the function of transmitting the plant coordinates recognized by the camera to the robotic arm, and the robotic arm moves to the target and then starts spraying the target.

Overall, during these two semesters, we completed most of the design and technical work. Although the function of the system is still a certain distance from our ideal function, we have basically achieved the task of automated spraying, so our project was successful. I think our project will affect most agricultural workers, especially greenhouse workers. Our products reduce the drift problem caused by large-scale pesticide spraying equipment. The saving of pesticides and automated design can greatly reduce agricultural costs and improve agricultural production efficiency. In addition, agricultural workers do not need to worry about being exposed to pesticides that are harmful to the body, and the damage to the surrounding environment is reduced.

References

- [1] M. Liu, J. Yuan, L. Song, Y. Zhang, & L. Liu, "Precise atomization uniform pesticide application control system and method for greenhouse," Chinese Patent CN111226892A, 2020. [Online]. Available: <https://patents.google.com/patent/CN111226892A/en>
- [2] X. Guo, Z. Zhou, S. Duan, X. Xia, G. Xu and W. Li, "Greenhouse spraying robot control system and control method," Chinese Patent CN115016476A, 2022. [Online]. Available: <https://patents.google.com/patent/CN115016476A/en>
- [3] X. Wang, Q. Feng, W. Ma, K. Jiang and P. Fan, " Automatic spraying robot for greenhouse," Chinese Patent CN103380766A, 2013. [Online]. Available: <https://patents.google.com/patent/CN103380766A/en>
- [4] Bagagiolo, Giorgia, et al. "Greenhouse Robots: Ultimate Solutions to Improve Automation in Protected Cropping Systems-A Review." MDPI, Multidisciplinary Digital Publishing Institute, 25 May 2022, www.mdpi.com/2071-1050/14/11/6436/htm.
- [5] Baltazar, André Rodrigues, et al. "Smarter Robotic Sprayer System for Precision Agriculture." MDPI, Multidisciplinary Digital Publishing Institute, 26 Aug. 2021, www.mdpi.com/2079-9292/10/17/2061.
- [6] Oberti, Roberto, and Ze’ev Schmilovitch. "Robotic Spraying for Precision Crop Protection." SpringerLink, Springer International Publishing, 1 Jan. 1970, link.springer.com/chapter/10.1007/978-3-030-77036-5_6.
- [7] Crop leaf disease detection system based on deep learning
wuxian.blog.csdn.net/article/details/129364610?spm=1001.2101.3001.6650.2&utm_medium=distribute.pc_relevant.none-task-blog-2~default~CTRLIST~Rate-2-129364610-blog-127673569.235%5Ev38%5Epc_relevant_anti_t3_base&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2~default~CTRLIST~Rate-2-129364610-blog-127673569.235%5Ev38%5Epc_relevant_anti_t3_base&utm_relevant_index=5&ydreferer=aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3NnY3Nkbjk5L2FydGljbGUvZGV0YWlscy8xMjc2NzM1Njk%3D. Accessed 8 Dec. 2023.
- [8] Dennis, Matthew. "How to Train a Custom YOLOv5 Model to Detect Objects." CodeProject, CodeProject, 25 Nov. 2022, www.codeproject.com/Articles/5347827/How-to-Train-a-Custom-YOLOv5-Model-to-Detect-Objec.

[9] Anwar, Aqeel. "What Is Average Precision in Object Detection & Localization Algorithms and How to Calculate It?" Medium, Towards Data Science, 13 May 2022, towardsdatascience.com/what-is-average-precision-in-object-detection-localization-algorithms-and-how-to-calculate-it-3f330efe697b.

[10] Experts in Spray Technology | Spraying Systems Co., www.spray.com/-/media/dam/industrial/usa/sales-material/catalog/cat75hyd_us_flat-spray_c.pdf. Accessed 9 Dec. 2023.

[11] Ozkan, Erdal. "Selecting the Right Type and Size of Nozzles for Effective Spraying in Orchards and Vineyards." Ohioline, 20 Dec. 2021, ohioline.osu.edu/factsheet/fabe-534.

[12] "Tech Library." Dultmeier Sales, www.dultmeier.com/technical-library/nozzle-volume-pressure.php. Accessed 8 Dec. 2023.

[13] Munson, Bruce Roy, 1940-. (2013). Fundamentals of fluid mechanics. Hoboken, NJ :John Wiley & Sons, Inc.,

[14] "Catalogs." KDLP1200 Diaphragm Pump Product Manual - Kamoer Fluid Tech (Shanghai) Co.,LTD. - PDF Catalogs | Technical Documentation | Brochure, pdf.directindustry.com/pdf/kamoer-fluid-tech-shanghai-co-ltd/kdlp1200-diaphragm-pump-product-manual/242598-1017455.html. Accessed 8 Dec. 2023.

Appendices

- Detailed data from the Beta prototype tests where the robotic arm targeted a single colored block and multiple colored blocks.

		the target position		
		90° X(m)	Y(m)	Z(m)
		Area(cm ²)		
one target				
1	0.306 0.752 0.092 63.825	0.055 0.092 66.798	0.055 0.092 66.798	0.055 0.092 66.798
2	0.306 0.752 0.091 63.825	0.055 0.092 66.833	0.055 0.092 66.833	0.055 0.092 66.833
3	0.307 0.752 0.091 63.897	0.056 0.092 68.756	0.056 0.092 68.756	0.056 0.092 68.756
4	0.307 0.752 0.091 63.897	0.056 0.092 68.756	0.056 0.092 68.756	0.056 0.092 68.756
5	0.306 0.749 0.091 63.884	0.055 0.092 68.641	0.055 0.092 68.641	0.055 0.092 68.641
6	0.306 0.749 0.091 63.884	0.055 0.092 68.641	0.055 0.092 68.641	0.055 0.092 68.641
7	0.306 0.748 0.092 67.151	0.056 0.092 66.798	0.056 0.092 66.798	0.056 0.092 66.798
8	0.306 0.748 0.092 67.151	0.056 0.092 66.841	0.056 0.092 66.841	0.056 0.092 66.841
9	0.306 0.748 0.092 67.151	0.056 0.092 66.841	0.056 0.092 66.841	0.056 0.092 66.841
10	0.306 0.748 0.092 67.151	0.056 0.092 66.841	0.056 0.092 66.841	0.056 0.092 66.841
11	0.307 0.747 0.092 65.143	0.055 0.092 65.394	0.055 0.092 65.394	0.055 0.092 65.394
12	0.307 0.747 0.092 65.143	0.055 0.092 65.394	0.055 0.092 65.394	0.055 0.092 65.394
13	0.307 0.752 0.091 63.807	0.056 0.092 65.735	0.056 0.092 65.735	0.056 0.092 65.735
14	0.307 0.752 0.091 63.807	0.056 0.092 65.735	0.056 0.092 65.735	0.056 0.092 65.735
15	0.307 0.753 0.092 64.216	0.056 0.092 66.261	0.056 0.092 66.261	0.056 0.092 66.261
16	0.307 0.753 0.092 64.216	0.056 0.092 66.261	0.056 0.092 66.261	0.056 0.092 66.261
17	0.306 0.749 0.092 65.683	0.055 0.092 65.394	0.055 0.092 65.394	0.055 0.092 65.394
18	0.306 0.749 0.092 65.683	0.055 0.092 65.394	0.055 0.092 65.394	0.055 0.092 65.394
19	0.306 0.749 0.092 65.683	0.055 0.092 65.394	0.055 0.092 65.394	0.055 0.092 65.394
20	0.306 0.747 0.092 67.449	0.055 0.092 68.399	0.055 0.092 68.399	0.055 0.092 68.399
21	0.306 0.751 0.092 66.807	0.057 0.092 70.8	0.057 0.092 70.8	0.057 0.092 70.8
22	0.306 0.751 0.092 66.807	0.057 0.092 70.8	0.057 0.092 70.8	0.057 0.092 70.8
23	0.306 0.751 0.092 66.807	0.057 0.092 70.8	0.057 0.092 70.8	0.057 0.092 70.8
24	0.306 0.751 0.092 66.807	0.057 0.092 70.8	0.057 0.092 70.8	0.057 0.092 70.8
25	0.306 0.751 0.092 66.807	0.057 0.092 70.8	0.057 0.092 70.8	0.057 0.092 70.8
26	0.306 0.751 0.092 66.807	0.057 0.092 70.8	0.057 0.092 70.8	0.057 0.092 70.8
27	0.306 0.751 0.092 66.807	0.057 0.092 70.8	0.057 0.092 70.8	0.057 0.092 70.8
28	0.306 0.751 0.092 66.807	0.057 0.092 70.8	0.057 0.092 70.8	0.057 0.092 70.8
29	0.309 0.752 0.092 67.484	0.056 0.092 67.586	0.056 0.092 67.586	0.056 0.092 67.586
30	0.309 0.752 0.092 67.484	0.056 0.092 67.586	0.056 0.092 67.586	0.056 0.092 67.586
31	0.305 0.75 0.092 64.211	0.054 0.092 66.621	0.054 0.092 66.621	0.054 0.092 66.621
32	0.305 0.75 0.092 64.211	0.054 0.092 66.621	0.054 0.092 66.621	0.054 0.092 66.621
Distance(m)	0.421088971	0.421088971	0.421088971	0.421088971
Gray Position(m)	0.1311493460474364	0.1311493460474364	0.1311493460474364	0.1311493460474364
Actual Position(m)	0.13 0.28 0.02	0.13 0.28 0.02	0.13 0.28 0.02	0.13 0.28 0.02
Broad(m)	0.007 0.003 0.002	0.001 0.002 0.001	0.001 0.002 0.001	0.001 0.002 0.001
two targets		the target position		
		90° X(m)	Y(m)	Z(m)
		Area(cm ²)		
1	0.364 0.713 0.091 47.746	0.059 0.091 43.336	0.059 0.091 43.336	0.059 0.091 43.336
2	0.364 0.758 0.091 39.168	0.059 0.091 43.336	0.059 0.091 43.336	0.059 0.091 43.336
3	0.364 0.748 0.092 39.168	0.059 0.091 43.336	0.059 0.091 43.336	0.059 0.091 43.336
4	0.364 0.758 0.092 43.118	0.059 0.091 43.336	0.059 0.091 43.336	0.059 0.091 43.336
5	0.364 0.758 0.092 43.118	0.059 0.091 43.336	0.059 0.091 43.336	0.059 0.091 43.336
6	0.364 0.758 0.092 43.118	0.059 0.091 43.336	0.059 0.091 43.336	0.059 0.091 43.336
7	0.364 0.758 0.092 43.118	0.059 0.091 43.336	0.059 0.091 43.336	0.059 0.091 43.336
8	0.363 0.753 0.098 41.849	0.058 0.092 41.849	0.058 0.092 41.849	0.058 0.092 41.849
9	0.363 0.753 0.098 41.849	0.058 0.092 41.849	0.058 0.092 41.849	0.058 0.092 41.849
10	0.363 0.753 0.098 41.849	0.058 0.092 41.849	0.058 0.092 41.849	0.058 0.092 41.849
11	0.363 0.753 0.098 41.849	0.058 0.092 41.849	0.058 0.092 41.849	0.058 0.092 41.849
12	0.363 0.756 0.091 42.434	0.058 0.092 41.849	0.058 0.092 41.849	0.058 0.092 41.849
13	0.364 0.758 0.091 40.829	0.058 0.092 41.849	0.058 0.092 41.849	0.058 0.092 41.849
14	0.364 0.758 0.091 40.829	0.058 0.092 41.849	0.058 0.092 41.849	0.058 0.092 41.849
15	0.364 0.758 0.091 40.829	0.058 0.092 41.849	0.058 0.092 41.849	0.058 0.092 41.849
16	0.364 0.758 0.091 40.829	0.058 0.092 41.849	0.058 0.092 41.849	0.058 0.092 41.849
17	0.363 0.753 0.098 41.849	0.058 0.092 41.849	0.058 0.092 41.849	0.058 0.092 41.849
18	0.363 0.758 0.091 42.434	0.058 0.092 40.666	0.058 0.092 40.666	0.058 0.092 40.666
19	0.359 0.751 0.091 42.396	0.058 0.092 40.666	0.058 0.092 40.666	0.058 0.092 40.666
20	0.364 0.758 0.091 42.118	0.058 0.092 40.666	0.058 0.092 40.666	0.058 0.092 40.666
21	0.364 0.758 0.091 42.118	0.058 0.092 40.666	0.058 0.092 40.666	0.058 0.092 40.666
22	0.364 0.758 0.091 42.118	0.058 0.092 40.666	0.058 0.092 40.666	0.058 0.092 40.666
23	0.364 0.758 0.091 42.118	0.058 0.092 40.666	0.058 0.092 40.666	0.058 0.092 40.666
24	0.364 0.758 0.091 42.118	0.058 0.092 40.666	0.058 0.092 40.666	0.058 0.092 40.666
25	0.364 0.758 0.091 42.118	0.058 0.092 40.666	0.058 0.092 40.666	0.058 0.092 40.666
26	0.364 0.758 0.091 42.118	0.058 0.092 40.666	0.058 0.092 40.666	0.058 0.092 40.666
27	0.364 0.758 0.091 42.118	0.058 0.092 40.666	0.058 0.092 40.666	0.058 0.092 40.666
28	0.364 0.758 0.091 42.118	0.058 0.092 40.666	0.058 0.092 40.666	0.058 0.092 40.666
29	0.364 0.758 0.091 41.857	0.058 0.092 40.666	0.058 0.092 40.666	0.058 0.092 40.666
30	0.364 0.758 0.091 41.857	0.058 0.092 40.666	0.058 0.092 40.666	0.058 0.092 40.666
31	0.364 0.758 0.091 41.714	0.058 0.092 40.666	0.058 0.092 40.666	0.058 0.092 40.666
32	0.364 0.758 0.091 41.714	0.058 0.092 40.666	0.058 0.092 40.666	0.058 0.092 40.666
The first one target				
Distance(m)	0.329387495	0.329387495	0.329387495	0.329387495
Gray Position(m)	0.1299719114301271	0.1299719114301271	0.1299719114301271	0.1299719114301271
Actual Position(m)	0.12 0.27 0.02	0.12 0.27 0.02	0.12 0.27 0.02	0.12 0.27 0.02
Broad(m)	0.000 0.002 0.001	0.000 0.002 0.001	0.000 0.002 0.001	0.000 0.002 0.001
The second one target				
Distance(m)	0.329387495	0.329387495	0.329387495	0.329387495
Gray Position(m)	0.09760233313790084	0.09760233313790084	0.09760233313790084	0.09760233313790084
Actual Position(m)	0.01 0.18 0.02	0.01 0.18 0.02	0.01 0.18 0.02	0.01 0.18 0.02
Broad(m)	0.003 0.002 0.002	0.003 0.002 0.002	0.003 0.002 0.002	0.003 0.002 0.002

- Detailed data from the Beta prototype tests where the robotic arm targeted a single real crop and multiple real crops.

One target	The target positions
	X(m), Y(m), Z(m), Area(cm^2)
1	0.191, 0.528, 0.047, 114.186
2	0.191, 0.528, 0.047, 114.186
3	0.19, 0.526, 0.047, 113.083
4	0.19, 0.526, 0.047, 113.083
5	0.191, 0.529, 0.047, 113.802
6	0.191, 0.529, 0.047, 113.802
7	0.191, 0.53, 0.047, 111.388
8	0.191, 0.53, 0.047, 111.388
9	0.191, 0.531, 0.047, 110.765
10	0.191, 0.531, 0.047, 110.765
Distance(m)	0.148445265
Spray Position(m)	0.10169521996310545, 0.28112605309172606, 0.025024478210816523
Actual Position(m)	0.1, 0.28, 0.02
Error(m)	0.001, 0.001, 0.005

Two targets	The target positions
	X(m), Y(m), Z(m), Area(cm^2)
1	0.254, 0.545, 0.047, 244.55
	0.254, 0.545, 0.047, 244.55
2	-0.052, 0.686, 0.049, 230.854
	0.254, 0.544, 0.047, 230.854
3	-0.052, 0.686, 0.049, 230.854
	0.254, 0.544, 0.047, 230.854
4	0.256, 0.561, 0.048, 228.046
	0.256, 0.561, 0.048, 228.046
5	-0.054, 0.692, 0.049, 231.873
	0.254, 0.55, 0.048, 231.873
6	-0.054, 0.692, 0.049, 231.873
	0.254, 0.55, 0.048, 231.873

7	0.255, 0.546, 0.047, 233.653
	0.255, 0.546, 0.047, 233.653
8	0.255, 0.545, 0.047, 246.79
	0.255, 0.545, 0.047, 246.79
9	-0.051, 0.685, 0.052, 249.362
	0.26, 0.564, 0.048, 249.362
10	-0.051, 0.685, 0.052, 249.362
	0.26, 0.564, 0.048, 249.362
11	-0.05, 0.682, 0.049, 232.221
	0.254, 0.552, 0.048, 232.221
12	-0.05, 0.682, 0.049, 232.221
	0.254, 0.552, 0.048, 232.221
The first one target	
Distance(m)	0.214939294
Spray Position(m)	0.12634366485694695, 0.27109172183872476, 0.023378552158568922
Actual Position(m)	0.13, 0.27, 0.02
Error(m)	0.004, 0.001, 0.003
The second one target	
Distance(m)	0.214939294
Spray Position(m)	-0.02261817532403405, 0.2983859283132185, 0.021313280593801318
Actual Position(m)	-0.2, 0.3, 0.02
Error(m)	0.002, 0.002, 0.001

Code part

1. Robotic arm part:

```

1 import serial
2 from interbotix_xs_modules.xs_robot.arm import InterbotixManipulatorXS
3 import numpy as np
4 import time
5 import os
6
7 def are_coordinates_equal(coord1, coord2, tolerance=0.05):
8     return all(abs(c1 - c2) <= tolerance for c1, c2 in zip(coord1, coord2))
9
10 def main():
11
12     ser = serial.Serial('/dev/ttyACM1', 9600)
13     # Replace 'COMX' with the appropriate COM port
14     time.sleep(5)
15
16
17     processed_coordinates = set() # Set to store processed coordinates
18
19     while True:
20         # Check if output.txt file exists and is not empty
21         if os.path.isfile('output.txt') and os.path.getsize('output.txt') > 0:
22             # Continue with robot arm operations
23             bot = InterbotixManipulatorXS(
24                 robot_model='px100',
25                 group_name='arm',
26                 gripper_name='gripper'
27             )
28
29             bot.arm.go_to_sleep_pose()
30
31             # Get the end effector position when the robot is in sleep pose
32             sleep_ee_pose = bot.arm.get_ee_pose()
33             translation1 = sleep_ee_pose[:3, 3]
34
35             bot.arm.go_to_home_pose()
36
37             # Get the end effector position when the robot is in home pose
38             home_ee_pose = bot.arm.get_ee_pose()
39             translation2 = home_ee_pose[:3, 3]
40
41             # Read xyz coordinates from output.txt and store them in a list of object poses
42             object_poses = []
43             with open('output.txt', 'r') as file:
44                 for line in file:
45                     try:
46                         object_pose = list(map(float, line.strip().split(' ', ' ')))
47                         area = object_pose[3]
48                         if area > 300:
49                             area = 300
50                         object_pose = [object_pose[0], object_pose[1], object_pose[2]]
51                         is_unique = True
52                         for processed_coord in processed_coordinates:
53                             if are_coordinates_equal(processed_coord, object_pose):
54                                 is_unique = False
55                                 break
56                         if is_unique:
57                             object_poses.append(object_pose)
58                             processed_coordinates.add(tuple(object_pose))
59                     except ValueError:
60                         print(f"Ignoring line in output.txt: {line.strip()}")

```

```

61
62         for i, object_pose in enumerate(object_poses, 1):
63             print(f"Object end effector pose {i} (under base coordinate system):\n", object_pose)
64
65             object_r = np.sqrt(object_pose[0]**2 + object_pose[1]**2 + object_pose[2]**2)
66             reach = 0.3
67             spraydistance = 0.01*(np.sqrt(4*area/np.pi))/0.8
68             print(f"disrance:\n", spraydistance)
69
70             if object_r > (reach + spraydistance):
71                 target_pose = [
72                     reach * object_pose[0] / object_r,
73                     reach * object_pose[1] / object_r,
74                     reach * object_pose[2] / object_r
75                 ]
76             else:
77                 target_pose = [
78                     (reach - spraydistance) * object_pose[0] / object_r,
79                     (reach - spraydistance) * object_pose[1] / object_r,
80                     (reach - spraydistance) * object_pose[2] / object_r
81                 ]
82
83             print("Target end effector pose (under base coordinate system):\n", target_pose)
84             print("Area:\n", area)
85
86
87             r = np.sqrt(target_pose[0]**2 + target_pose[1]**2)
88
89             bot.arm.set_single_joint_position(joint_name='waist', position=np.round(np.arctan2(target_pose[1], target_pose[0]), 3))
90
91             bot.arm.set_ee_cartesian_trajectory(x=(-translation2[0] + r), z=(-translation2[2] + target_pose[2]))
92
93
94
95             # final_ee_pose = bot.arm.get_ee_pose()
96             # print(f"End effector pose {i} (under base coordinate system):\n", np.round(final_ee_pose, 2))
97
98             time.sleep(6)
99
100            final_ee_pose = bot.arm.get_ee_pose()
101            print(f"Final end effector pose {i} (under base coordinate system):\n", np.round(final_ee_pose, 2))
102
103            spray_ee_pose = bot.arm.get_ee_pose()
104
105            spray_pose = spray_ee_pose[:3, 3]
106
107            reference_pose = [object_pose[0] - spray_pose[0], object_pose[1] - spray_pose[1], object_pose[2] - spray_pose[2]]
108
109
110            alpha = -np.abs(np.round(np.arctan2(reference_pose[1], np.sqrt(reference_pose[0]**2 + reference_pose[1]**2)), 3))
111
112            bot.arm.set_single_joint_position(joint_name='wrist_angle', position=1.5*alpha)
113            print()
114
115
116            ser.write(b'S')  # Sending the signal to start spraying
117

```

```
118
119
120         time.sleep(8)
121         bot.arm.go_to_home_pose()
122         time.sleep(3)
123
124         # Calculate the target position and perform actions accordingly
125
126         # Your actions for each object_pose go here...
127
128         bot.arm.go_to_home_pose()
129         bot.arm.go_to_sleep_pose()
130
131         # Clear output.txt after processing
132         open('output.txt', 'w').close()
133
134         # Exit the program after processing
135         break
136     else:
137         # Wait for output.txt file to have content
138         time.sleep(1)
139
140 if __name__ == '__main__':
141     main()
142
```

2. Camera:

```

1 import pyrealsense2 as rs
2 import numpy as np
3 import cv2
4 import random
5 import torch
6 import time
7 import shutil
8 import serial
9
10 import platform
11 import pathlib
12 plt = platform.system()
13 if plt != 'Windows':
14     pathlib.WindowsPath = pathlib.PosixPath
15
16
17 model = torch.hub.load('ultralytics/yolov5', 'custom', r'/home/qiuhaoyolov5-D435i-main/crop_best.pt')
18 model.conf = 0.5
19
20 def save_matrix_to_txt(matrix, filename):
21     with open(filename, "a") as file: # 使用追加模式打开文件
22         matrix_as_string = ', '.join(map(str, matrix))
23         file.write(matrix_as_string + "\n")
24
25
26 # Function to copy file to another directory
27 def copy_file(source_file, destination_directory):
28     shutil.copy(source_file, destination_directory)
29
30 def get_mid_pos(frame, box, depth_data, randnum):
31     distance_list = []
32     mid_pos = [(box[0] + box[2])//2, (box[1] + box[3])//2] #确定索引/深度的中心像素位置
33     min_val = min(abs(box[2] - box[0]), abs(box[3] - box[1])) #确定深度搜索范围
34     #print(box)
35     for i in range(randnum):
36         bias = random.randint(-min_val//4, min_val//4)
37         dist = depth_data[int(mid_pos[1] + bias), int(mid_pos[0] + bias)]
38         cv2.circle(frame, (int(mid_pos[0] + bias), int(mid_pos[1] + bias)), 4, (255,0,0), -1)
39         #print(int(mid_pos[1] + bias), int(mid_pos[0] + bias))
40         if dist:
41             distance_list.append(dist)
42     distance_list = np.array(distance_list)
43     distance_list = np.sort(distance_list)[randnum//2-randnum//4:randnum//2+randnum//4] #冒泡排序+中值滤波
44     #print(distance_list, np.mean(distance_list))
45     return np.mean(distance_list)
46
47 def dectshow(org_img, boxes, depth_data, depth_intrin):
48     img = org_img.copy()
49
50     for box in boxes:
51         cv2.rectangle(img, (int(box[0]), int(box[1])), (int(box[2]), int(box[3])), (0, 255, 0), 2)
52         dist = get_mid_pos(org_img, box, depth_data, 24)
53
54         cv2.putText(img, box[-1] + str(dist / 1000)[:6] + 'm', (int(box[0]), int(box[1])), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2)
55         a = int(box[0])
56         b = int(box[1])

```

```

57     x1 = int(box[0])
58     y1 = int(box[1])
59     x2 = int(box[2])
60     y2 = int(box[3])
61
62     cv2.circle(img, (x1, y1), 5, (0, 0, 255), -1) # 红色圆点
63     cv2.circle(img, (x2, y1), 5, (0, 255, 255), -1) # 黄色圆点
64     cv2.circle(img, (x2, y2), 5, (255, 0, 0), -1) # 蓝色圆点
65
66     depth1 = aligned_depth_frame.get_distance(x1, y1) # 获取第一个点的深度值
67     depth2 = aligned_depth_frame.get_distance(x2, y1) # 获取第二个点的深度值
68     depth3 = aligned_depth_frame.get_distance(x2, y2) # 获取第三个点的深度值
69
70     point1 = rs.rs2_deproject_pixel_to_point(depth_intrin, [x1, y1], depth1)
71     point2 = rs.rs2_deproject_pixel_to_point(depth_intrin, [x2, y1], depth2)
72     point3 = rs.rs2_deproject_pixel_to_point(depth_intrin, [x2, y2], depth3)
73
74     # 计算两点之间的距离
75     width = np.linalg.norm(np.array(point2) - np.array(point1))
76     height = np.linalg.norm(np.array(point3) - np.array(point2))
77     Area = width * height * 10000
78     # print("宽 is:", width, "m")
79     # print("长 is:", height, "m")
80
81     cv2.putText(img, f'Area: {str(Area)[:8]}cm^2', (int(box[0]), int(box[1]) - 40),
82                 cv2.FONT_HERSHEY_SIMPLEX, 0.5,
83                 (255, 255, 0), 2)
84     cv2.putText(img, f'W: {str(width)[:4]}m H: {str(height)[:4]}m', (int(box[0]), int(box[1]) - 25),
85                 cv2.FONT_HERSHEY_SIMPLEX, 0.5,
86                 (0, 255, 0), 2)
87
88
89
90
91 for i in range(len(boxs)):
92     X = int((boxs[i][2] + boxs[i][0]) / 2)
93     Y = int((boxs[i][3] + boxs[i][1]) / 2)
94     dis = aligned_depth_frame.get_distance(X, Y)
95     camera_xyz = rs.rs2_deproject_pixel_to_point(depth_intrin, (X, Y), dis)
96     camera_xyz = np.round(np.array(camera_xyz), 3)
97     cv2.circle(img, (X, Y), 4, (255, 255, 255), 5)
98
99     # cv2.putText(img, str(camera_xyz), (X,Y), 0, 1,[0, 0, 255], thickness=2, lineType=cv2.LINE_AA)
100    # print('%.f,%.f,%.f',camera_xyz[0],camera_xyz[1],camera_xyz[2])
101
102
103
104    # theta = np.pi/2
105    theta = -np.pi / 2 if n >= stepmotortime/2 else np.pi / 2
106
107    h = 0.04
108    d = 0.24
109    matrix1 = np.array([[1, 0, 0, d], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]])
110    matrix2 = np.array([[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, h], [0, 0, 0, 1]])
111    matrix3 = np.array([[np.cos(-np.pi/2), -np.sin(-np.pi/2), 0, 0], [np.sin(-np.pi/2), np.cos(-np.pi/2), 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]])
112    matrix4 = np.array([[1, 0, 0, 0], [0, np.cos(-np.pi/2), -np.sin(-np.pi/2), 0], [0, np.sin(-np.pi/2), np.cos(-np.pi/2), 0], [0, 0, 0, 1]])

```

```

114 matrix5 = np.array([[np.cos(theta), 0, np.sin(theta), 0], [0, 1, 0, 0], [-np.sin(theta), 0, np.cos(theta), 0], [0, 0, 0, 1]])
115
116
117
118 # Perform matrix multiplication using np.dot()
119 result_mat = matrix1 @ matrix2 @ matrix3 @ matrix4 @ matrix5
120
121
122 cam = np.array([[camera_xyz[0]], [camera_xyz[1]], [camera_xyz[2]], [1]])
123
124 cam_final = np.dot(result_mat, cam)
125
126
127 result = [cam_final[0,0], cam_final[1,0], cam_final[2,0]]
128
129 # coordinates_detected = False
130
131
132 result = [cam_final[0, 0], cam_final[1, 0], cam_final[2, 0], Area]
133
134
135
136
137 if result[1] == 0:
138     continue # Skip this coordinate if it has long decimal values or y-value is 0
139
140 # Round the coordinates to 3 decimal places
141 result = tuple(round(coord, 3) for coord in result)
142
143 # coordinates_detected = True
144
145
146 save_matrix_to_txt(result, "output.txt")
147
148
149
150
151
152 # Specify the source file (output.txt) and destination directory
153 source_file = "output.txt"
154 destination_directory = "/home/qiuhaoo/Design/output.txt"
155
156
157 # Call the function to copy the file
158 copy_file(source_file, destination_directory)
159
160 # with open("output.txt", "w") as file:
161 #     file.write(result)
162 print(camera_xyz)
163
164 #cv2.circle(img, (640, 360), 8, [0, 0, 255], thickness=-1)
#cv2.imshow('dec_img', img)

```

```

167     if __name__ == "__main__":
168         # Configure depth and color streams
169         pipeline = rs.pipeline()
170         config = rs.config()
171         config.enable_stream(rs.stream.depth, 640, 480, rs.format.z16, 30)
172         config.enable_stream(rs.stream.color, 640, 480, rs.format.bgr8, 30)
173         # Start streaming
174         pipeline.start(config)
175         align_to = rs.stream.color # 与color流对齐
176         align = rs.align(align_to)
177         n = 0
178         stepmotortime = 8
179
180
181
182
183     ser = serial.Serial('/dev/ttyACM1', 9600) # Replace '/dev/ttyUSBX' with the correct serial port
184     time.sleep(3)
185     ser.write(b'M')
186
187
188     try:
189         while True:
190             # Wait for a coherent pair of frames: depth and color
191             frames = pipeline.wait_for_frames()
192             depth_frame = frames.get_depth_frame()
193             color_frame = frames.get_color_frame()
194             #新添内容
195             aligned_frames = align.process(frames)
196             aligned_depth_frame = aligned_frames.get_depth_frame()
197             depth_intrin = aligned_depth_frame.profile.as_video_stream_profile().intrinsics |
198
199             if not depth_frame or not color_frame:
200                 continue
201             # Convert images to numpy arrays
202
203             depth_image = np.asarray(depth_frame.get_data())
204
205             color_image = np.asarray(color_frame.get_data())
206
207             results = model(color_image)
208             boxes= results.pandas().xyxy[0].values
209
210             #boxes = np.load('temp.npy',allow_pickle=True)
211             dectshow(color_image, boxes, depth_image, depth_intrin)
212             camera_xyz = dectshow(color_image, boxes, depth_image, depth_intrin)
213             # Apply colormap on depth image (image must be converted to 8-bit per pixel first)
214             depth_colormap = cv2.applyColorMap(cv2.convertScaleAbs(depth_image, alpha=0.03), cv2.COLORMAP_JET)
215             # Stack both images horizontally
216             images = np.hstack((color_image, depth_colormap))
217             # Show images
218             #cv2.namedWindow('RealSense', cv2.WINDOW_AUTOSIZE)
219             #cv2.imshow('RealSense', images)
220             key = cv2.waitKey(1)
221             start_time = time.time()
222             # Press esc or 'q' to close the image window
223
224             time.sleep(1)
225             n = n + 1
226
227             if key & 0xFF == ord('q') or key == 27 or n == steplmotortime:
228                 cv2.destroyAllWindows()
229                 break
230
231
232     finally:
233         # Stop streaming
234         pipeline.stop()
235         ser.close()
236
237

```

3. Stepper motor and spray system:

```

1  #include <Arduino.h>
2  #include <LiquidCrystal_I2C.h>
3  #include <PID_v1.h>
4  #include <AccelStepper.h>
5
6  const int pwmPin = 9;
7  const unsigned int desiredFrequency = 10000;
8
9  volatile int flow_frequency;
10 float vol = 0.0, l_minute;
11
12 unsigned char flowsensor = 2;
13 unsigned long currentTime;
14 unsigned long cloopTime;
15
16 double Setpoint;
17 double Input;
18 double Output;
19 double Kp = 1, Ki = 0.01, Kd = 10;
20
21 #define STEPS_PER_REVOLUTION 200
22 #define STEP_PIN 4
23 #define DIR_PIN 3
24 #define ENABLE_PIN 7
25
26 AccelStepper stepper(AccelStepper::DRIVER, STEP_PIN, DIR_PIN);
27
28 PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);
29
30 LiquidCrystal_I2C lcd(0x27, 16, 2);
31
32 void setupTimer2() {
33   TCCR2A = _BV(COM2A1) | _BV(WGM20) | _BV(WGM21);
34   TCCR2B = (TCCR2B & B11111000) | B00000010; // Added parentheses
35   OCR2A = (16000000 / 8 / desiredFrequency) - 1;
36 }
37
38 void flow() {
39   flow_frequency++;
40 }
41
42 void rotateDegrees(float degrees); // Function prototype
43
44 void setup() {
45   pinMode(pwmPin, OUTPUT);
46   Serial.begin(9600);
47   setupTimer2();
48   pinMode(flowsensor, INPUT);
49   digitalWrite(flowsensor, HIGH);
50   attachInterrupt(digitalPinToInterrupt(flowsensor), flow, RISING);
51   Serial.begin(9600);
52   lcd.init();
53   lcd.clear();
54   lcd.backlight();
55   lcd.setCursor(0, 0);
56   lcd.print("Water Flow Meter");
57   lcd.setCursor(0, 1);
58   lcd.print("Circuit Digest");
59   currentTime = millis();
60   cloopTime = currentTime;

```

```

61   Input = 0;
62   myPID.SetMode(AUTOMATIC);
63   myPID.SetTunings(Kp, Ki, Kd);
64   Serial.begin(9600);
65   pinMode(ENABLE_PIN, OUTPUT);
66   digitalWrite(ENABLE_PIN, LOW);
67   stepper.setMaxSpeed(1000);
68   stepper.setAcceleration(500);
69 }
70
71 void loop() {
72   currentTime = millis();
73   if (currentTime >= (cloopTime + 1000)) {
74     cloopTime = currentTime;
75     if (flow_frequency != 0) {
76       l_minute = (flow_frequency / 98.00);
77       lcd.clear();
78       lcd.setCursor(0, 0);
79       lcd.print("Rate: ");
80       lcd.print(l_minute);
81       lcd.print(" L/M");
82       l_minute = l_minute / 60;
83       lcd.setCursor(0, 1);
84       vol = vol + l_minute;
85       lcd.print("Vol:");
86       lcd.print(vol);
87       lcd.print(" L");
88       flow_frequency = 0;
89       Serial.print(l_minute, DEC);
90       Serial.println(" L/Min");
91     } else {
92       Serial.println(" flow rate = 0 ");
93       lcd.clear();
94       lcd.setCursor(0, 0);
95       lcd.print("Rate: ");
96       lcd.print(flow_frequency);
97       lcd.print(" L/M");
98       lcd.setCursor(0, 1);
99       lcd.print("Vol:");
100      lcd.print(vol);
101      lcd.print(" L");
102    }
103  }
104
105  if (Serial.available() > 0) {
106    char signal = Serial.read();
107
108    if (signal == 'M') {
109      rotateDegrees(-90);
110      delay(5000);
111      rotateDegrees(90);
112      delay(1000);
113      rotateDegrees(90);
114      delay(5000);
115      rotateDegrees(-90);
116      delay(1000);
117      Serial.println(" okkkkk");
118    }

```

```
119     if (signal == 'S') {
120         Setpoint = 200;
121         myPID.Compute();
122         analogWrite(pwmPin, output);
123         delay(5000);
124         analogWrite(pwmPin, 0);
125     }
126 }
127 }
128 }
129
130 void rotateDegrees(float degrees) {
131     long targetPosition = stepper.currentPosition() + (degrees * STEPS_PER_REVOLUTION / 360);
132     stepper.moveTo(targetPosition);
133     while (stepper.distanceToGo() != 0) {
134         stepper.run();
135         delay(20);
136     }
137 }
138 }
```