

---

# AUTOMATIC AGRICULTURAL SOFT FRUIT HARVESTING ROBOT

---

PROJ 515

PLYMOUTH UNIVERSITY, U.K.

{KHAIAN.MARSH, JAKE.SHAW-SUTTON, ISAAC.CHASTEAU }@STUDENTS.PLYMOUTH.AC.UK

-Student Number-

Khaian Marsh 10475727

Jake Shaw-Sutton:10424696

Isaac Chasteau: 10453090

## **I. Abstract**

This report covers the research, design, prototyping and programming taken place to construct a 1 meter long automatic strawberry picking robotic arm.

This system utilises a desynchronized stereo vision camera set-up to produce a 3-dimensional point cloud map of its environment. From here, colour analysis is performed across both the HSV and RGB colour spaces to produce a dual compared target map of all the matched regions along with a cartesian estimation on size and location. The system is then able to determine which order the identified objects can be inspected, and then produces a path, void of any obstacles, to then intersect with the target object. Once in contact, the gripping mechanism produces a low-pressure region between the fruit and its deformable nozzle, after which, the camera unit is then pivoted down and a structured lighting rig triggered so that the ripeness of the fruit can be analysed.

If ripe, the fruit is manipulated away from the plant by the arm, before the head of the unit then is driven forwards enclosing the fruit. The cutting mechanism is then driven, closing around the stem of the fruit, cutting it free from the plant. The fruit is then transported whilst safely contained within the head of the end-effector to a storage container.

The arm pitch and yaw is actuated by 2 Nema 23 stepper motors with 15:1 Planetary gearboxes [1] and a single Nema 23 stepper motor [2] is used to drive the Aluminium 1 inch box section [3] with m1 steel rack [4]. These are operated by modified ST-M5045 2-Phase stepper drivers [5], controlled by an Arduino Nano microcontroller [6]. This microcontroller also uses an I2C (Inter-Integrated Circuit) communication [7] to control a 12-bit 16 channel PCA9685 servo driver [8] which provides PWM (Pulse Width Modulation) [9] signals to operate the connected SG90 9G [10] and HJ S3315D [11] Servo motors. This system is controlled through a USB Serial connection [12] of an ASUS TinkerBoard Microcomputer [13] with the CANYON CNE-CWC2 Web cameras [14] also directly connected.

## **2. Acknowledgements**

The authors would like to thank Dr Zaki Ahmed, for supervising the project, giving advice on the project management, and organizing guest lecturers to speak.

The authors would also like to thank Sheila Storm, John Eastment, and Kevin Allan for dealing with components orders, as well as the rest of the Smeaton 303 technical staff for keeping the laboratory running.

The authors would like to give a final thank you to Dr Martin Stoelen, for providing his advice about soft robotics and soft fruit manipulation.

### **3. Table of Contents**

<b>1. Abstract .....</b>	<b>1</b>
<b>2. Acknowledgements.....</b>	<b>2</b>
<b>3. Table of Contents .....</b>	<b>3</b>
<b>4. Glossary.....</b>	<b>7</b>
<b>5. Equations.....</b>	<b>8</b>
<b>6. Introduction.....</b>	<b>9</b>
<b>7. Specifications.....</b>	<b>10</b>
<b>8. Project Management.....</b>	<b>11</b>
8.1. Project Team.....	11
8.2 Team Management .....	11
8.3 Initial Research .....	12
8.4 CAD Design Planning.....	12
8.5 Software Planning.....	13
<b>9. Research .....</b>	<b>14</b>
9.1 Existing Prototypes and Methods .....	14
9.1.1 Dogtooth Technologies .....	15
9.1.2 Manual Labour .....	16
9.2 Environmental Investigation .....	17
9.2.1 Kitley Farm .....	17
9.3 Strawberry Manipulation .....	18
9.4 Ripeness evaluation .....	18
9.5 Microcontrollers/Microprocessors .....	19
9.6 Servo Motor Comparisons .....	21
9.7 Rotary Motor Comparisons .....	23
9.8 Stepper Motor Comparisons .....	23
9.9 Machine Vision.....	24
9.9.1 The Need for Stereo Cameras .....	24
9.9.2 The Chosen Cameras .....	25
9.9.3 Justification for Chosen Cameras .....	25

9.9.4 Camera Calibration.....	<b>26</b>
9.9.5 Stereo Camera Orientation .....	<b>27</b>
9.9.6 Stereo Vision Depth Mapping .....	<b>28</b>
9.10 GPS Positioning .....	<b>29</b>
9.11 Construction Methods .....	<b>29</b>
9.11.1 3D Printing .....	<b>30</b>
9.11.1 Laser Welding.....	<b>31</b>
9.11.1 Lost Casting .....	<b>31</b>
9.12 Material Research .....	<b>32</b>
9.12.1 PLA .....	<b>32</b>
9.12.2 ABS .....	<b>32</b>
9.12.3 Flex 45 .....	<b>33</b>
9.12.4 Aluminium Extrusion .....	<b>33</b>
<b>10. Theory.....</b>	<b>36</b>
10.1 I2C Communication .....	<b>36</b>
10.2 Coordinate Frames.....	<b>37</b>
10.3 Relating the Camera Coordinate Frame to the Base Coordinate Frame .....	<b>38</b>
10.4 Kinematics.....	<b>40</b>
10.5 PWM Servo Control.....	<b>42</b>
10.6 DC Cutting Motor .....	<b>43</b>
10.7 Lens Distortion .....	<b>43</b>
10.8 Calculating Coordinates Relative to the Camera .....	<b>45</b>
10.9 Controlling the Relay .....	<b>45</b>
10.10 Limit Switches .....	<b>45</b>
10.11 Simultaneous Stepper Motor Movements .....	<b>46</b>
10.12 Colour Spaces.....	<b>47</b>
10.13 Path Planning .....	<b>47</b>

10.14 Object Identification .....	48
10.15 Ripeness Estimation .....	50
10.16 Strawberry Identification .....	53
10.17 Algorithm Detection Testing .....	56
10.18 Stereo Distance Estimation .....	57
<b>11. Prototype Design.....</b>	<b>59</b>
11.1 Arm .....	59
11.2 End Effector .....	61
11.2.1 Grasping Methods.....	61
11.2.2 Solid Gripper Designs .....	61
11.2.2.1 Clamp Grasp .....	61
11.2.2.2 Enclosure Design .....	62
11.2.2.3 Advantages and Disadvantages .....	62
11.2.3 Deformable Gripper Designs .....	62
11.2.3.1 Deforming Fingers .....	63
11.2.3.2 Stem Cutting.....	63
11.2.3.3 Advantages and Disadvantages .....	63
11.2.4 Vacuum Testing.....	64
11.2.4.1 Contact Surfaces.....	64
11.2.4.2 Fan Design .....	66
11.2.5 Combining Elements .....	67
11.2.6 Structured Lighting.....	67
11.2.7 Camera Positioning .....	69
11.2.8 End Effector Manipulation .....	70
11.2.8.1 Dynamic Displacement Control .....	70
11.2.8.2 Design Prototype .....	70
<b>12. Final Prototype Design .....</b>	<b>72</b>

<b>13. Programming.....</b>	<b>73</b>
13.1 Programming Overview .....	73
13.2 Code Overview .....	73
13.3 Mode Average Algorithm.....	77
13.4 Stereo Matching Algorithm.....	77
13.5 Storing Positions .....	77
13.6 Serial Communications.....	78
13.7 Stepper Control.....	80
<b>14. Testing .....</b>	<b>84</b>
14.1 HJ S3315D Servo Testing .....	84
14.2 Micro Servo Testing .....	85
14.3 Linear Movement Speed .....	86
14.4 Arm Maximum Payload.....	86
<b>15. Future Developments.....</b>	<b>88</b>
15.1 Drive Improvements .....	88
15.2 Circuitry Improvements .....	89
15.3 Limit Switch Improvements .....	89
15.4 Platform Improvements .....	89
15.5 Camera Improvements.....	90
15.6 Algorithm Improvements .....	90
15.7 End Effector Improvements .....	91
<b>16. Conclusion .....</b>	<b>92</b>
16.1 Expenditure Break-Down .....	93
16.2 Team Dynamic .....	94
<b>17. References.....</b>	<b>95</b>
<b>18. Appendices.....</b>	<b>100</b>

## **4. Glossary**

PLA - Polylactic Acid

LED - Light Emitting Diode

I2C - Inter-integrated Circuit

SDA - Serial Data line

SCL - Serial Clock line

SIL - Single In Line

Hz - Hertz (One cycle per second)

PCB - Printed Circuit Board

ADC - Analog to Digital Converter

DOF - Degrees Of Freedom

CAD - Computer Aided Design

USB - Universal Serial Bus

PWM - Pulse Width Modulation

FDM - Fused Deposition Modelling

SD Card - Secure Digital Card

OS - Operating System

HSV - Hue Saturation Value

RGB - Red Green Blue

SI Units - International System of Units

## **5. Equations**

This section covers some of the important equations that were used and the derivation of the inverse kinematic equations.

Ohms law:

$$V = I \cdot R$$

Where:  $V$  is voltage in Volts,  $I$  is current in Amps and  $R$  is resistance in Ohms

Pythagoras's Theorem:

$$a^2 + b^2 = c^2$$

Trigonometric equations:

$$\sin \theta = \frac{Opp}{Adj} \quad \cos \theta = \frac{Adj}{Hyp} \quad \tan \theta = \frac{Opp}{Adj}$$

Cosine Rule:

$$\cos A = \frac{b^2 + c^2 - a^2}{2 \cdot b \cdot c} \quad \cos A = \frac{a^2 + c^2 - b^2}{2 \cdot a \cdot c} \quad \cos A = \frac{a^2 + b^2 - c^2}{2 \cdot a \cdot b}$$

Torque:

$$\tau = rF \cdot \sin \theta$$

Where:  $\tau$  is torque in  $N \cdot m$ ,  $r$  is the position vector ( $m$ ),  $F$  is the force vector ( $N$ )  
and  $\theta$  is the angle between the force vector and lever vector

## **6. Introduction**

Back in 1927, the global population just exceeded 2 billion people [15] and since then, it has continuously risen at an alarming rate. In 2017, the human population of earth now exceeds 7.5 billion people [16] and the united nations department of economic and social affairs predicts that by the year 2050, this figure will have increased up to 9.7 billion [17] (Fig. 1). Already in cities and countries across the globe, there are signs of overpopulation [18] and many industries are being put under continuous pressure to deliver more to help sustain this growing population. One industry that is vital to the health and well-being of that increasing population, and as a result, has the largest challenge to tackle is the agricultural industry.

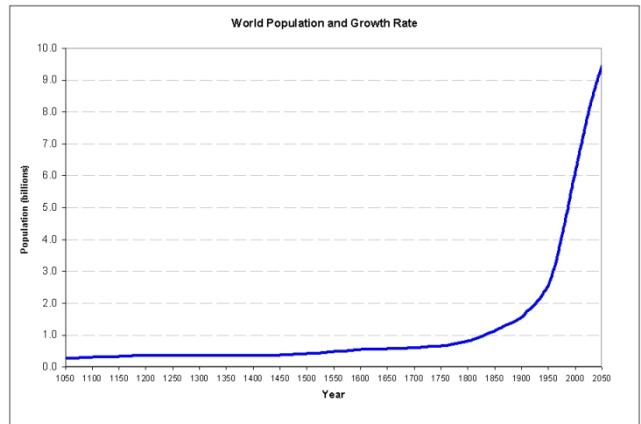


Fig. 1 : Graph Showing the World Population between the years 1050 – 2050. [26]

Farming has remained the same for countless years with many roles being sustained by human labour, however, this has not continued. There has been a severe lack of willing labour [19] to fulfil these roles and it is reaching a limit where new methods need to be utilised. Not only are consumers demanding higher quality produce in larger volumes, they want these made without the use of pesticides or other chemicals that can harm the environment [20]. As a result, the number of farms that have been unable to maintain a successful harvest has increased [21].

Many aspects of the agricultural world have been able to embrace technology, boosting performance and raising the boundaries of production to ever increasing levels. Autonomous tractors aiding in grain collection [22], automatic grape harvesters [23]. Where ever there is a possibility for improvement, farming has embraced the technology to benefit it. However, there is one aspect that technology has yet to tackle and that is the solid collection of soft produce.

Soft produce collection is still a manual process. Many workers must be employed to man the isles of vegetation, carefully picking or plucking each fruit. This is because many of these items are too delicate to mechanically grasp without becoming bruised or damaged [24]. Alongside this, the challenge of identifying the goods from their environment is too unreliable and unpredictable.

This project proposes that with soft robotic implementation, in addition to a smart machine vision system, such a challenge is possible. It was decided that the issue surrounding the

harvesting of the Genus Fragaria species (aka. strawberries) [25] would be attempted, as it is a fruit that is cultivated worldwide.

Constructing and developing such a robotic system to harvest said fruit, can be divided out to 3 main aspects. The first aspect is to how such a fruit can be grasped and removed from the plant without subjecting it to cosmetic damage or physical harm such as bruising. Causing such damage would result in a fruit that would be less marketable to the final consumer and as such undesirable. Once grasped the fruit would then need to be safely transported to a place of storage. This leads into the second aspect of such a system. A mechanism that can move the fruit grasping end effector to and from its initial and final destinations in a manner that allows for the avoidance of objects within its path. Having such flexibility in transportation would mean that collisions with the plant or other fruit could be avoided. Finally, such a robot would require some way of automatically identifying the location of a ripe fruit and calculating the most appropriate path to move the end effector along. Only with these 3 systems working in harmony would such a task be viable.

## **7. Specifications**

This is a simplified specification outline determining the base principals this project must achieve to complete the desired task. This was set out to ensure that each team member was aware of the other systems that are required for this project to be successful.

### ***The final system must:***

- Identify fruit directly upon the plant
  - o Machine Vision with a colour depth map
- Plot a movement path that does not damage the surrounding environment
  - o Obstacle avoidance
- Grasp and manipulate the fruit without causing adverse harm to them
  - o Soft robotic interfacing
- Analyse the fruit and determine their ripeness
  - o Methods of fruit ripeness evaluation
- Remove the fruit from its plantation without damaging the vegetation
  - o Will the stem need to be cut or not?
- Store the fruit for transportation
  - o Any special requirements?

## **8. Project Management**

This project was independently undertaken as part of PROJ 515. The intent of this project was to provide a technical challenge to each of the members of the group and produce a functional system prototype. The final system was broken down into 3 main stages with each member assigned as a head for each stage of development. This allowed for all three members of the team to be working in parallel on adjacent aspects rather than time being lost transitioning one stage between each of the members.

A Gantt chart (Fig. 2) was also produced with an initial set time frame with milestones for each member across the scope of this project. This Gantt chart was produced alongside a contingency plan to ensure development of the project was not halted due to unforeseen circumstances affecting any member of the team.



Fig. 2: Initial Project Development Gantt Chart

### **8.1 Project Team**

The team consists of 3 members, each of which was studying on the MEng Robotics course at Plymouth University. As such, each member shares similar experience and could aid each other in feedback and guidance.

### **8.2 Team Management**

This project was divided out to three main sections and each team member was put in charge of a respective system.

The division of the sections can be seen in Table 1.

Team Members	Stage of Development
Jake Shaw-Sutton	End-Effector – Stage 1
Khaian Marsh	Arm – Stage 2
Isaac Chasteau	Vision Processing – Stage 3

Table 1: Team members with their associated development phase

During individual development, each member had to specialise and coherently work together to produce a functional prototype within the allocated time span that was available. For that reason, it was very important for each team member to continuously maintain clear communication and contact with each other member. So that any issues that might have risen could have been foreseen and handled.

Over the course of the project, each team member would update their fellow members with a bi-weekly meeting held either in person or over an online conference call via Skype [27], depending which option was more convenient. During which, each member spoke about the progress that they had made over the previous time span and the current challenges that they were facing. This allowed for suggestions to be put forward to help aid in one another's developments. In addition, an updated Gantt chart was maintained each week so that the progress could be checked to ensure the final target was still on track.

### 8.3 Initial Research

Before this project was undertaken, a wide understanding of the current market and applicable technologies was required. This would ensure that any development created, would not just be a repeat of work that has been completed previously. As such, an IP landscape (often referred to as a Patent Landscape) document was produced. This project consists of developing a physical working prototype unit, and as a result, by analysing existing patent documents, a design can be produced that not only is capable of fulfilling its allocated task, but it can avoid infringement of pre-existing concepts so that if the project were to be carried forwards, the design would be completely independent. This would allow for both future development and investment opportunities.

### 8.4 CAD Design Planning

As this is a prototype unit that is being constructed by a team rather than a single individual, initial resources were put into place to help aid in cross development and communication. For example, it was decided that this project would utilise the platform of Autodesk Fusion 360 [28] with its shared cloud project management. This provides up to date access to all of the design files directly for every member. As this is a cloud based system, it means that when each member saved a design, it was then automatically uploaded, allowing for the rest of the team to view and suggest alterations or improvements immediately.

This resource also ensured that any member of the group was able to then select a file and begin 3D printing it, this meant that even if a team member did not have access to a 3D printer, or theirs was in use producing another part, a request could be formally forwarded and the new part could enter the manufacturing process.

## 8.5 Software Planning

In addition to the resource put in place for CAD files, a similar system was used for the projects programming. A GitHub repository [28] was created which allowed each member direct access to the latest saved version of the projects code.

Implemented directly upon the windows file system, every time a file is opened and saved, the new file is automatically uploaded with the previous version being archived. This meant that when a feature was implemented and it directly affected the performance of the rest of the system, the code version could be reverted, ensuring such issues could be easily avoided.

## **9. Research**

This section of the document outlines many of the factors that were taken into consideration and researched before any direct design work began on the prototype. By ensuring that a rich knowledge base on the task at hand is shared by all of the members of the group, it allows for foreseeable issues to be spotted early and enables a continuous development process to be sustained.

### **9.1 Existing Prototypes and Methods**

Researching existing products, both in commercial use and research development use, there are very few solutions to autonomously harvest strawberries. The majority of these solutions focus on simplifying the task of picking strawberries, by structuring the environment in which they are presented to the robot.

This is performed by growing the strawberries in raised beds (Fig. 3) which have black tarpaulin wrapped around the surface of the soil, and the planter. By maintaining a high and narrow plantation, the growing strawberries can be hung over the edge of the planter, resulting in the fruits stem orientation to be more uniform and in stark contrast to the colour of the object behind it. As such, this reduces the complexity of both the computer vision software, and the hardware required to pick them.

While this is a viable strategy, it is not compatible with existing strawberry farms which have the strawberries planted in fields on the ground, and the construction of specialised planting boxes, may make the deployment of such a system prohibitively expensive. The key is to develop a system that, rather than specialising for one specific growing pattern, can generalise and be adapted to many different growing patterns, all dependant on what the farm is currently using.

Most of these designs also tend to be deployed in poly-tunnels. The reason for this is to further structure the environment that the fruits are growing in. It prevents wind from dynamically altering the fruits relative spatial location and diffuses the direct light from the sun to reduce strong shadows. They also aid in further weather protection not just to the plants, but also to the prototypes being tested as many of the designs are early in development and as such, are not weather proof.



Fig. 3: Ozaki Laboratory of Utsunomiya University strawberry picking prototype picking from controlled plantation bedding. [30]

### 9.1.1 Dogtooth Technologies [31]

A start-up company based in Cambridge, Dogtooth Technologies aims to build “smart autonomous robots for harvesting soft fruits such as strawberries.” [32]. They have produced a prototype unit that can identify and pick strawberries that are over hanging the edges of raised plantation beds. (Fig. 4).



Fig. 4: Dogtooth Technologies Prototype working on raised plantations [31]

The system they have developed consists of a pair of solid, narrow gripper fingers with a stereo camera mounted directly below, housed within the wrist. The cameras are able to locate the fruit based upon a colour matching depth map and the gripper is used to then grasp the stem of the fruit, cutting it, to then move it from the plant to a storage hopper.

The advantage of this system is that the control algorithms for identifying the fruit can be kept relatively simple, with the cameras then able to provide further information regarding the fruit once it has been picked. For example, after picking and before deposition for transport, the robot can perform quality control checks and then grade the fruit without the need of repeated handling. It can also gather data regarding the plant as a whole. This includes information such as the yield from each plant, enabling precision crop management to be performed by the farmer to maintain successful future harvests.

The disadvantage of this system is that a restricted plantation set-up is required for the robot to effectively work. Plants must be raised high upon beds that are above the working height of the arm, and only fruit that hangs over from the plants can be picked. If the stem of the fruit is not clearly visible to the cameras located upon the head of the arm, then it cannot be picked.

This is only a prototype unit, however the current iteration poses quite strong limitations to the future development of it. Strawberries can grow in any location on the plant and in any orientation. This means that a large number of fruits will be un-attainable by it, requiring additional manual picking to make up for this shortfall.

## 9.1.2 Manual Labour

The main method of currently harvesting strawberries is by hand. Groups of workers attend each alley of plantation in the field and slowly work their way from one end to the other. Additionally, It is advised that strawberries are cut half an inch up from the stem by applying pressure between your thumbnail and index finger [33]. However there has been increasing worry from farmers regarding the use of manual labour.

With the news that Great Britain will be departing from the European union, there have been increasing concerns over the decline in available foreign workers for such low paid manual roles [34] that the fruit farming industry relies upon heavily. This insecurity has been growing recently, as eastern European countries have continued developing and their local jobs are becoming more appealing to their residence. Now that Brexit has been announced there is further concern over how importing labour will work with the biggest suspected issue regarding growing paper work and costs. This all in turn may lead foreign workers to work elsewhere unless offered a higher wage, which for farmers may not be possible due to further financial pressures. This would mean, fewer, more expensive workers.

Ultimately, the concept of designing a system that can perfectly handle any situation is very difficult. For the task of picking strawberries, even if the machine is unable to pick every fruit, it will still dramatically reduce the number left to be manually sorted. Rather than requiring dozens of people working in the fields each day, with the implementation of a competent automated system, this number can be brought down, until only one person is required to follow the robot, picking only the fruit it is unable to collect.

## 9.2 Environmental Investigation

When designing a prototype or developing a robot to be placed into an environment, it is vital to have a clear understanding of what challenges it may face. This is relevant not just to the task that it is performing, but also to the surroundings that it will have to work within. As such, the team headed out to a local produce farm for first-hand experience of what strawberry farming entails.

### 9.2.1 Kitley Farm

Located in Yealmpton near Plymouth, there is a family business located upon the site of Kitley Farm called “PIPS pick your own” [35]. Here there are various arrays of fruits and vegetables grown which are all available for visitors to pick themselves. Among other things, this includes carrots, runner beans, and most importantly, strawberries.

The strawberries grown here are grown in the most common manner as found nationwide in strawberry growing facilities. They are laid out across a field in columns, with gaps of roughly 1500mm between each row for pickers to use and walk down. In addition, the ground is covered in a black glossy plastic with holes that each plant grows through (Fig. 5). This is to prevent weeds from growing in between the plants and reducing the quality of the field’s soil.

From this visit, various aspects surrounding the nature of strawberry picking were learnt and a further insight into the task was revealed. In addition, it further emphasized the need for such a system as, even for a commercial unit that allows for individuals to pick their own produce, large volumes are still picked and sorted by manual labourers. It was also discovered that the ideal time to pick such fruit is at night whilst the temperature is lower, then having the fruit immediately frozen or chilled as this prolongs the fresh life span of the produce. Lastly, the fruit can grow in any orientation upon the plant, often in clusters. As such, this must be taken into consideration for the functional design. A unit that can manipulate and sever the fruit from any location upon the plant without requiring a specific orientation.



Fig. 5: Strawberry growing area of Kitley Farm

### 9.3 Strawberry Manipulation

Strawberries are a soft fruit that can grow in a multitude of sizes and shapes (Fig. 6), however they still easily bruise from impacts upon their surface. Ideally, they require a soft material that can de-form to absorb any pressure placed upon them and distribute it evenly. It was learnt during the visit to the strawberry farm, that when picking strawberries, the main body of the fruit should not be handled. Instead they should be grasped by the stem and then cut. However, once they have been cut, strawberries are later sorted by hand to determine what grade they are for selling. If this grading can be performed at the same time that they are removed from the plant, this would reduce the total number of times that they are handled. This would also reduce the time between the strawberry being removed from the plant, and being on the retail shelf, which in turn would result in a longer shelf life. Such a system would even allow for customers to order their fruit in advance, with it being freshly picked specifically for them.



Fig. 6: Grade 1, same species, Strawberry size comparison

Another factor that restricts how strawberries can be manipulated is their degradation when placed within a volume of water. If strawberries were transported inside bodies of water, this would provide ample protection from shocks and impacts. However, the issue is that their surface then degrades at an accelerated rate, resulting in a reduced shelf life. This means, the chosen system cannot use any form of lubrication to move the fruit and instead, must rely solely upon a grasping method.

### 9.4 Ripeness evaluation

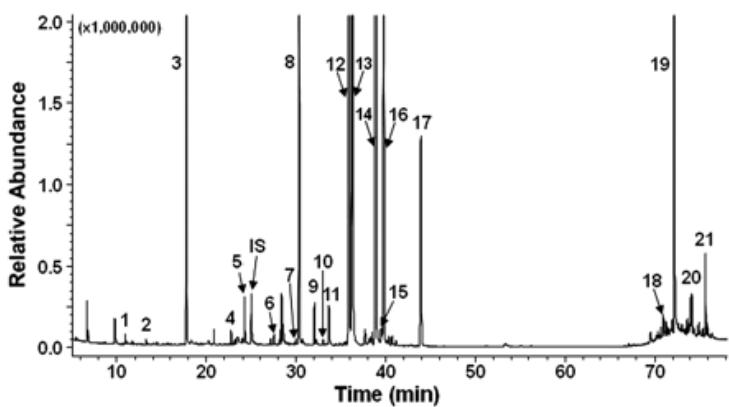
When picking strawberries, it is important to only pick the fruit that have fully matured and are ready for eating. If they are picked whilst they are still developing, this hinders the plants capability to produce more.

There are many factors that can be used to determine the ripeness of the fruit. The most common one is the colour of the exterior surface. When under-developed, the skin of the strawberry maintains a white appearance, from this it slowly transitions between green and finally a red. The transition of this colour occurs from the tip of the strawberry, up to the stem. The area around the

stem is the last region that matures and as such, this can be used to determine the ripeness. If around the stem of the fruit, there is a large white region, then it has not developed enough.

One method for detecting this change would be the use of colour sensors or cameras to identify the percentage of the surface of the fruit that has fully transitioned. If this percentage is high enough, then it can be assumed to be ripe. If it is detected to be below a certain threshold, then it can be assumed to not be ripe, and left on the plant.

Another way a strawberry's ripeness can be determined is by their internal metabolic composition (Fig. 7). When ripe, significant changes in their metabolite levels can be detected and this alters the internal structure of the fruit. This can be detected using ultrasonic transducers, by analysing their reactance to certain frequencies that the compounds resist. The disadvantage to analysing the fruit with this method is that a good transition medium is required between the transducer and the fruit to allow for accurate readings. Such a medium would require an application to every fruit, before then some way of cleaning them. Because water can speed up the degrading process, it was deemed that this method for ripeness detection should not be used.



**Fig. 7:** GC-MS analysis on polar metabolites from strawberry fruits (TIC). (1) Butanedioic acid; (2) 2,5-dimethyl-4-hydroxy-3(2H)-furanone; (3) malic acid; (4) arabinonic acid; (5) gluconic acid; (6) mannonic acid; (7) 2-keto-D-gluconic acid; (8) citric acid; (9) 2-hydroxyethylsulphonic acid; (10) gulonic acid; (11) glucose; (12 and 13) fructose; (14 and 16) galactose; (15) palmitic acid; (17) *myo*-inositol-1-phosphate; (18) hexadecanoic acid; (19) sucrose; (20) turanose; (21) palatinose; IS, internal standard. See reference [36]

## 9.5 Microcontrollers/Microprocessors

As this project entails the construction of a working robotic arm, there will need to be an appropriate microcontroller implemented to handle signal processing and communication between the proto-types actuators. This will need to talk to a microprocessor that is powerful enough to real-time process a stereo camera feed and perform kinematic calculations. This next section compares a various assortment of appropriate development boards that suit the required criteria.

## Arduino Nano [6]

Running at a clock speed of 16MHz and utilising an Atmel ATmega328 chipset, the 5volt compliant Arduino Nano (Fig. 8) is a very popular option among makers as there are a huge assortment of prebuilt libraries and functions that cover almost any interaction.



Fig. 8: The Arduino Nano [6]

## Atmel ATTiny [37]

5 volt compliant with a built-in oscillator of 8MHz, the Atmel ATTiny product range (Fig. 9) is an array of dual in-line and surface mount chips. They require only power lines, with no additional supporting circuitry required to allow them to work as fully fledged microcontrollers. However, they all share very limited flash memory sizes, 8Kb, meaning that only small programs can be uploaded and run on them.

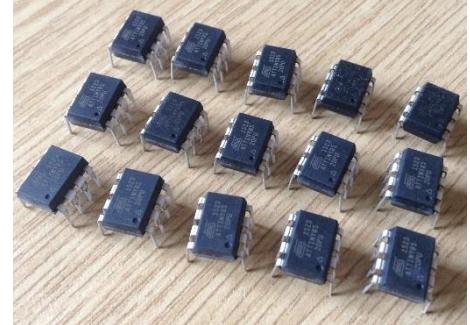


Fig. 9: The Atmel ATTiny 85, 8-pin dip package

## Raspberry Pi 3 Model B [38]

The Raspberry pi 3 model B (Fig. 10) is a 1.2GHz quad-core micro-computer with 1GB of available RAM and 4 USB ports. This is another very popular option with makers as the 40 GPIO pins allow for easy custom communication between different modules whilst still running a fully open source operating system.



Fig. 10: The Raspberry Pi 3 Model B [38]

## ASUS Tinkerboard [13]

The ASUS TinkerBoard (Fig. 11) is a new quad-core 1.8GHz single-board computer with a separate 600MHz GPU and 2GB of dual channel DDR3 Ram. This micro-computer also has a 40 pin GPIO header and shares the same footprint design as the raspberry pi.



Fig. 11: The ASUS TinkerBoard [13]

When comparing the micro-controllers for this project, it was important to take into consideration future expandability and adaptation. As such, with the limited memory size of the ATTiny microcontrollers, it was found that using an Arduino Nano would be an appropriate choice as it not only shares a small footprint, but has much more program memory and GPIO pins for control. This means that there are more possibilities with what can be performed by the Nano compared to any of the ATTiny range.

Comparing the micro-computers, the main focus was on their ability to perform advanced image processing and analysis tasks. With both a higher clock speed, and a dedicated GPU, the ASUS TinkerBoard stood out ahead of the Pi for its processing power. The Raspberry Pi also suffers from having a slow speed USB splitter. This means that the USB ports all share the same data lines from the CPU and instead the data is re-distributed by a secondary on-board chip. The ASUS TinkerBoard however has direct access from the CPU to the USB ports resulting in a higher data transfer rate. This is very beneficial as most cameras use a USB interface, so by having 2 high speed interfaces compared to the Pi's single, the TinkerBoard should be more capable of receiving the input camera data feed.

## 9.6 Servo Motor Comparisons

Part of this project will require accurately driven actuation for multiple joints, and the most common method for achieving this would be by utilising servo motors. With a closed loop feed-back system that uses a potentiometer to measure the current driven angle, and a geared DC motor to drive the shaft, servo motors can offer both high torque and precision. This section compares 4 common yet different types that would be suitable for this prototype.

### SG90 Micro Servo [10]

With a plastic gearbox and a rated torque of 1.8Kg/cm, the 9g SG90 micro servo (Fig 12) is an ideal choice for actuation of small joints. The plastic gears do hold a high probability of sheering; however, this servo motor is best suited to minimal force operations. Other more expensive metal geared variants do exist, such as the MG90S [39], however these do not provide an increase in rated torque, only an increase in gear tooth strength.



Fig. 12: SG90 Micro Servo [10]

## Futaba S310 Standard Servo [40]

Seen as one of the most popular and common hobby servo motors, the Futaba S310 (Fig. 13) has plastic gears and a rated torque of 6Kg/cm. This has become the benchmark to grade any standard sized servo motor as the design and dimensions have become universal.



Fig. 13: Futaba S310 [40]

## HJ S3315D Servo Motor [11]

With a metal gear box and a rated torque of 16Kg/cm, the HJ S3315D (Fig. 14) is a very powerful, and very affordable hobby servo motor. The main disadvantage is that it does not have the same communication control capability as a similar strength digital servo motor.



Fig. 14: HJ S3315D Servo Motor

## Dynamixel AX-12A Robot Actuator [41]

Claimed to be the most advanced robotic servo on the market, the Dynamixel range of servos (Fig. 15) have become very popular in robotic research. With plastic gears and a stall torque of 15Kg/cm, they do not quite match the output power of the HJ S3315D range of servos, however they make up for that factor by their communication protocol. Digitally interfaceable through a one wire interface, they are very easy to communicate with and daisy chain together.



Fig. 15: Dynamixel AX-12A Robot Actuator [41]

Comparing the attributes of these 4 servo motors, it became clear that depending on the scenario, each servo has a beneficial factor. For example the dynamixels are great for linking together, reducing the total wiring required, however the HJ servo motors are more suitable when more torque is simply required. For this project, the main comparability factor was their price. This project was intended to be built in as low cost a manner as possible, and as a result, the HJ S3315D and SG90 were the most suitable options. They cover both possible requirements of a small size for on-board joint actuation, and a high torque for larger joint actuation.

## 9.7 Rotary Motor Comparisons

This project required a very small high torque motor to actuate the final cutting mechanism design. For this, there was only one motor that was found applicable and that was an unbranded DC 300RPM 6V Metal Geared Motor (Fig 16). These types of motors are very common, all of them with slightly different attributes, however this provided more than enough torque than was required to fulfil its allocated task.



Fig. 16: DC 300RPM 6V Metal Geared Motor

## 9.8 Stepper Motor Comparisons

As the arm wants to have as long as reach as possible to account for various planting arrangements, it was decided to use a one meter long floating box for the linear section. As an approximation, the arm will be assumed to be one meter long, allowing for the length of an end effector and that the linear section will not be able to extend to a full meter.

Assuming a worst-case scenario where the arm is horizontal at full extent:

- The end effector will be made of PLA, and will mostly use light weight servos. It can be assumed to weight approximately 500 grams and be a meter from the pivot
- The box section assumed to be a 20mm square box with 3mm wall thickness weighing approximately 300 grams with a COM (centre of mass) 500mm from the pivot.
- The steel gear rack weighing 1100 grams and with a COM 500mm from the pivot
- The weight of the strawberry can be ignored

Item	Weight (g)	Distance between COM and pivot (mm)	Torque generated (Nm)
End effector	500	1000	4.905
Box section	300	500	1.4715
Gear rack	1100	500	5.3955
Total:			<b>11.772</b>

Table 2: Calculating theoretical weight of the arm

This gives a worst-case scenario torque of 11.7 Nm (see Table 2). As the arm wants to be able to move as fast as possible, this should be achieved using as little gearing as possible, without inducing too much weight.

For this reason, a NEMA 23 stepper with a 15:1 planetary gearbox was chosen, this setup gives 12.7Nm of torque. This gives the pitch joint sufficient torque and speed, for almost all of the arms workspace, when driven by a micro stepping driver and only weights 1.8Kg.

The motor should be able to drive the arm at its full extend, all be it at a reduced speed, this was considered preferable compared to the overall speed reduction of the next available gearbox (47:1).

## 9.9 Machine Vision

### 9.9.1 The Need for Stereo Cameras

To pick the strawberries their position in space must be known relative to the robot arm so that it can move to them. To identify strawberries without needing contact with them a camera is needed. A camera alone would not give depth information about the location, so another sensor is needed.

The main types of distance sensors are laser rangefinders, ultrasonic sensors, light coding systems (Kinect V1), time-of-flight cameras (Kinect V2), and stereo cameras.

Laser rangefinders (Fig. 17) only give the distance of a single point, and often have a relatively long minimum range; of upwards of 1m. As the arms workspace has approximately a 1m radius, that would not give useable distance information, and a single point makes obstacle avoidance difficult.



Fig. 17: Laser Range Finder [52]

Ultrasonic sensors (Fig. 18) have a minimum range upwards of 44mm which would allow targets to be within the workspace. However due to the angle of the emitted sound wave, it would not be possible to ascertain the distance of a single target strawberry on a plant.



Fig. 18: HC-SR04 Ultrasonic Sensor

Light coding systems like the Kinect V1 (Fig. 19) can have the infrared (IR) projected pattern washed out by the IR in sunlight. As the robot is designed to pick strawberries in a field, there will often be bright direct sunlight that will mask the pattern, rendering the sensor useless. The Kinect V2 (Fig. 20) uses the time-of-flight of projected IR grid which suffers the same problem in bright sunlight.



Fig. 19: Kinect V1 [53]



Fig. 20: Kinect V2

Stereo cameras (Fig. 21) overcome all of these issues. They have high enough resolution to detect individual strawberries and will give accurate depth information up to approximately 150mm away dependent upon the distance and angle between the cameras.



Fig. 21: Stereo Camera Example

### 9.9.2 The Chosen Cameras

The cameras used for the computer vision aspect of the project were two Canyon HD (Fig. 22) Web Cameras (model number CNE-CWC2). They have a maximum framerate of 30 frames per second (fps), at a resolution of 640x480 pixels. CMOS image sensor technology is used in the cameras; they are manual focus, although auto white balance is implemented on them.



Fig. 22: Chosen Camera [54]

### 9.9.3 Justification for Chosen Cameras

The web cameras have a universal serial bus (USB) interface which makes them simple to connect to and use, with both standard computers and the Asus Tinker board. The USB web cameras' video stream can easily be obtained in code using the OpenCV libraries that are available for both Windows, and Linux distribution.

The cameras had to be fitted into some form of stereo bracket to keep them in the same position, relative to one another, for the stereo matching algorithms to work properly. This to make this compact, the casing of the cameras would need to be removed and the PCBs mounted directly into the bracket. This would require the internal circuitry to be relatively small.

Since the cameras are separate, they may not be synchronised, which could cause issues with stereo matching; if there is movement in the field of view, between when the two cameras capture the stereo image. This was not predicted to cause a significant problem because the imaged could be captured while the arm was not moving and the strawberry plants would only move due to wind. Unless the wind is particularly strong the larger, ripe strawberries that would be detected should not move significantly so would not cause problems when calculating their position. The cameras have a button to initiate the capture of a snapshot or video. These buttons could be wired together and used to synchronize the stereo image capturing.

The frame rate of the cameras is not important for the matching algorithm, and the image processing will slow the rate of captured frames, thus making it the limiting factor, not the frame rate of the cameras.

The camera resolution does not need to be particularly high, as the targets are strawberries, and will need to be relatively close to the cameras so that the arm may reach them. This means that strawberries in range will never be too small to be seen by the cameras. A benefit of using lower resolution is that image processing will be faster, and result in a higher frame rate.

For the stereo matching algorithms to work the cameras need to be calibrated and not move relative to each other after that. If the focus of one of the cameras changes after calibration it will invalidate the generated calibration matrices, preventing the matching algorithm from performing well. Having manual focus allows it to be set before calibration and not change afterwards.

Automatic white balance does not affect the calibration of the cameras however if the cameras change too drastically from each other then it may be hard to match pixels between them. As both cameras are mounted next to each other and will be looking at the same scene then they should both adjust their white balances almost simultaneously and to the same levels.

The cameras are cheap compared to other stereo cameras, costing only £9.98 each. The bracket is small and 3D printed so costs a negligible amount. The Intel RealSense costs upwards of £100 depending on the distributor and the Kinect V2 with the required adapters costs just under £95, making this stereo camera setup approximately five times cheaper.

## 9.9.4 Camera Calibration

Camera calibration was performed using black and white chessboard pattern, with 26mm squares and that has 9x6 black-white corner intersections. The pattern was printed off and attached to a 3mm thick aluminum plate to ensure it remained flat and without distortion during the calibration process.

Initially calibration and testing was done with the camera mounted horizontally next to each other as shown in figure 23. A visual display of the calibration algorithm is shown in figure 24 which shows a stereo image pair of the calibration pattern attached to the aluminum plate and the detected corners circled.



Fig 23: Initial Stereo Camera

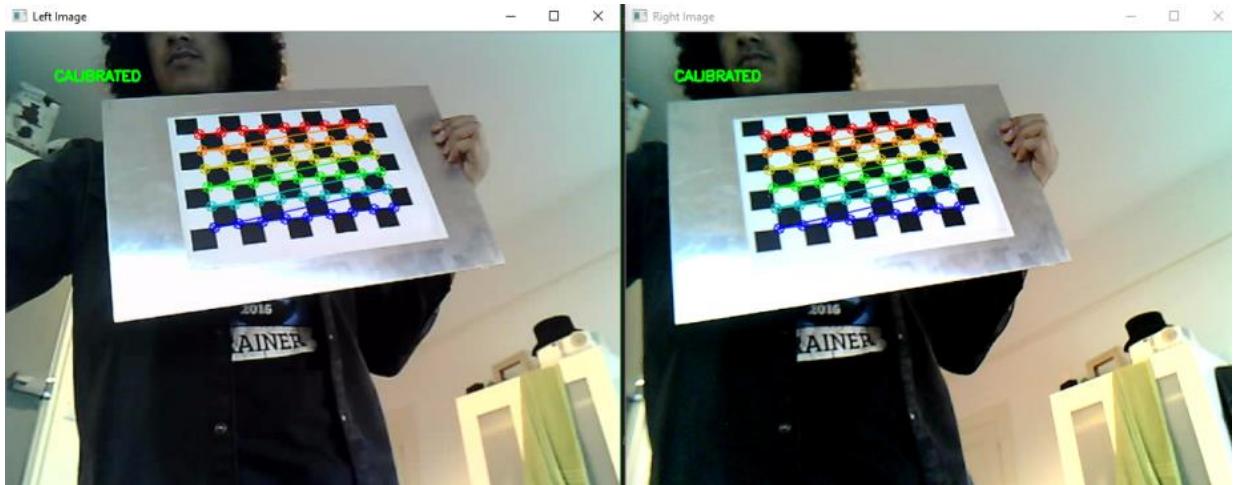


Fig 24: Stereo Calibration

To give more control over the calibration process the same code was written to save stereo image pairs and another code sample was used to perform the calibration from these saved images. Calibrating from saved images allows for poor quality image pairs to be selectively deleted and for images causing an undesired distortion in the final calibration to be deleted.

For the final calibration, the cameras were mounted into the stereo camera bracket and used to capture stereo image pairs of the calibration pattern. 21 stereo pairs were used to produce the best calibration results as shown in figure 25.

### 9.9.5 Stereo Camera Orientation

The two cameras were initially mounted horizontally next to each other, then calibrated and tested. The distance between the two camera lenses (baseline) of this configuration was 40.6 mm. It was determined that accurate disparity values could only be reliably obtained at distances greater than about 30mm from the cameras. To make the footprint of the stereo camera bracket smaller the cameras would need to be mounted one above the other as shown in figure 26. This gives the stereo camera a baseline of 25.6mm and allows for the disparity of closer objects to be calculated up to approximately 130mm away from the cameras. When the cameras were put in this orientation with both of their top edges aligned, the stereo image's top and bottom edges did not align well and resulted in a large portion being cropped. This was caused by the slight angle the lenses were mounted onto the PCB. To give a larger field of view, one of the cameras was turned around so that its bottom edge aligned with the top

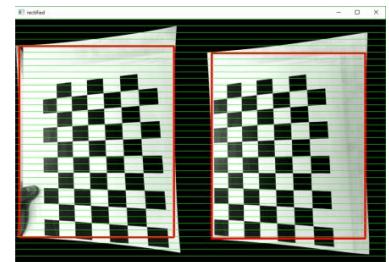


Fig 25: Final Calibration



Fig 26: Camera Orientation

edge of the other camera. As both lenses were mounted at approximately the same angle, the stereo images aligned more accurately, giving the desired field of view.

## 9.9.6 Stereo Vision Depth Mapping

Stereo vision is based on the principle of obtaining two different views of the same 3D scene and comparing the matching points in them to give a disparity map. The disparity map is inversely proportional to the relative distance at the pixel coordinates.

To achieve the pixel matching, both images must match the projection of an ideal pinhole camera. This requires removing the lens distortion from both images and reprojecting them onto the same plane.

Calibration using the OpenCV libraries calculates both the cameras intrinsic and extrinsic parameters and the associated matrices. These are then used to undistort and reproject the images onto a common plane.

To reduce the search space for match the pixels epipolar geometry is used, this reduces the search space from 2D to 1D.

Figure 27 shows the focal length of the cameras  $f$ , the distance between the cameras (baseline)  $b$ , the image centers  $(x_0, y_0)$  and  $(x'_0, y'_0)$ , and the distance  $z$  to point  $p$ . Disparity is given by the equation:

$$\text{disparity} = x' - x$$

Both images are aligned when they are reprojected so there is no disparity in the  $y$  axis.

The accuracy of the disparity value is dependent on the baseline, relative distance, resolution of the cameras and accuracy of the calibrations. A larger baseline will increase the accuracy of the disparity but it also increase the difficulty with matching points in the images. This is due to the scene captured having being more different.

Due to the way disparity is calculated it necessitates that the maximum disparity is larger than  $x' - x$ , therefore:

$$x > x' - \text{maxDisparity}$$

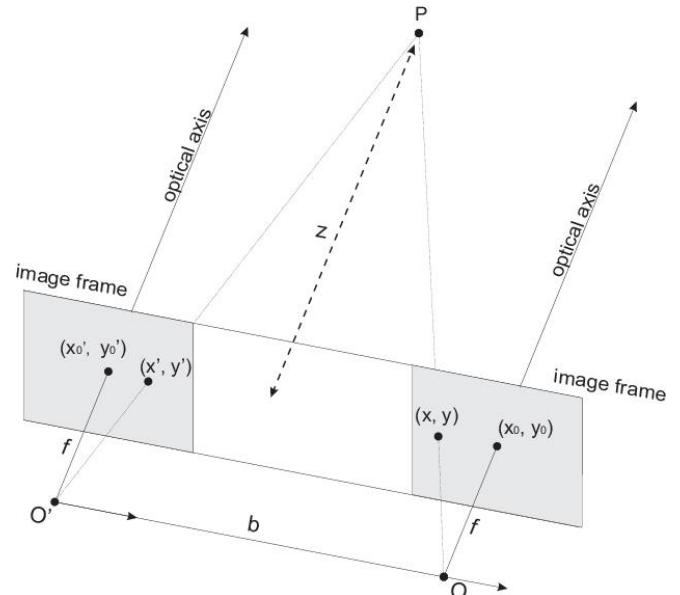


Fig 27: Stereo Coplanar Cameras [50]

This means that these cannot be any stereo correspondence with the two images between the first columns of the left image up to the value of maxDisparity. As can be seen in figure 28, the left columns are blank.

## 9.10 GPS Positioning

One of the big challenges with implementing a moving platform, is that of navigation. Guiding the robot between plants is a relatively easy task for computer vision and/or dead reckoning. The bigger issue is guiding the robot over longer distances. Either to move the robot between rows or fields or to allow the robot to move to a station to recharge, or a point to drop-off the fruits it has harvested. While the task of building a working base is not in the specifications for this project, it is worth planning for as it is a necessity for a complete system.



Fig 28: Left Disparity Column

One solution to this problem is using the GPS or the Russian GLONASS to provide location data, using their respective satellite networks, both systems cover all the latitudes that strawberries are commonly grown. The issue with these systems is that they require a RTK (Real Time Kinematic) mast at a known position to correct for the various inaccuracies that the systems naturally have. These masts are becoming increasingly common in the farming industry for guiding semi-autonomous tractors and combine harvesters, as they can provide a position accurate to the centimetre. If such a system were to be used it would allow the robot to be used effectively anywhere, with relatively little setup.

For large scale farms that can use the system all year, the £500 to £750 yearly subscription fee is generally considered acceptable. For our system, that has a harvesting window of, at best, 5 months (May-September in the UK), and is generally on a lot smaller scale, such a fee would be harder to justify. A further study would be needed to determine if this yearly fee would be viable or if an alternative solution would be needed.

## 9.11 Construction Methods

As this project requires building a large solution to a difficult task, there was no single construction method that would have been able to achieve all the desired design requirements. As such, this project contains a mixture of hand crafted wood and metal elements, alongside rapidly prototyped CAD parts.

### 9.11.1 3D Printing

There are two common methods of 3D printing, FDM (Fused Deposition Modelling) and SLA (Stereo lithography); only the first method will be covered in this report. An example of an FDM Desktop 3D printer can be seen in Fig. 29.

FDM printing is when a spool of material, often known as filament, is melted and then ejected out of a nozzle which itself is being moved along the X, Y and Z coordinates relative to the shape of the item being produced. Standard 3D FDM printers create the model layer by layer slowly working up towards the top of the model. This is a very useful tool as it allows for very complex and intricate shapes to be designed and then printed from the virtual world. However, there are some properties that are often over looked when 3d printing is being discussed as a means of rapid prototyping.



Fig. 29: Desktop 3D printer used for this project.

The first is orientation of an object. If you are to design any object or item to be 3D printed then extra consideration must be used to determine what orientation the object needs to be printed from. This is because as the print head works up, it is constantly extruding material and that material is affected by gravity. If there are any over hangs on an object or design then appropriate support material needs to be generated so to allow for said sections to be extruded in the air. It is important to note too that these supports can use large amounts of extra filament so getting the right orientation of an object will also result in the reduction of waste material.

The second fact is also related to orientation but instead affects the physical properties of the designed piece. Whenever printing an object, the practical use of said object needs to be considered. For example, if you are creating a bracket which will be experiencing a lateral amount of force when in use, the object will want to be printed so that the force is distributed along the length of a layer and not along the direction between layers. If force is exerted between layers then the resulting piece will be very fragile and prone to snapping between the two layers experiencing the most force. When force is applied along a layer's length, the strain is distributed on all of the joining layers and much greater forces are able to be imparted. It also should be mentioned that when designing to distribute forces around an object, any right angles will want to be smoothed into a gradual gradient, so that any force imparted is distributed over a slowly changing area rather than a

direct intersection as this will then act as the weak failing point of the design if performed incorrectly.

### 9.11.2 Laser Welding

This is a method that is useful for the flat surface joining of two plastic parts. Laser welding involves using a high-power hand-held laser, and melting the profiles of two printed parts together. This produces a material bond along where the laser melts the plastic. Such methods are very useful for CAD parts as it allows for the objects to be digitally split. The multiple pieces can then be printed in the most appropriate orientation depending on their force requirement. Without this technique, joints would have to be fixed in place, usually by nuts and bolts which increases the minimum part size to accommodate them.

### 9.11.3 Lost Casting

One manufacturing method that was considered was lost casting. This entails making a sacrificial part out of a soft material and packing it in a sand mould.

Molten aluminium is then poured into the mould, turning the part into gas and forming a replica of it. This method allows parts to be designed in an easily worked material before turning it into a high strength copy. Traditionally a wax or a low-density foam is used as the sacrificial material, but with 3D printers it is possible to use PLA, this effectively allows 3D printed parts to be turned into high strength aluminium castings, quickly and without the need of complex machining.

The problem with lost casting is that it is not an exact science and can fail or distort without a professional setup. Lost casting also produces solid parts, which goes against the objective of keeping the arm as light as possible.

Ultimately there was no situation in which a part needed the strength and complexity that could be provided by lost casting, so the process was not used on any final design part of the unit.

## 9.12 Material Research

When it comes to 3D printing parts, there are now a wide range of materials that can be chosen between all with their own unique advantages and disadvantages. These range from Bamboo lined filament which allows for the final item to be sanded and stained in the same fashion as a hand crafted wooden piece can be. To carbon lined filament which is then able to print conductive tracks. The most common materials to be found being used by a 3D FDM printer however are Polylactic Acid (PLA) and Acrylonitrile Butadiene Styrene (ABS). However recently there have been an increase in the number of available flexible filaments. Materials that once printed, retain a high strength and flexibility property. One such example is Flex 45 [42].

### 9.12.1 PLA

PLA is a biodegradable plastic which is made from corn starch and as a result is a renewable resource. When being used by a 3D printer, it has a very low chance of warping as it does not contract when it cools down. However over long periods of time or when imparted with large forces it can also deform and lose its shape.

### 9.12.2 ABS

ABS is a petroleum based plastic and is commonly found in many consumer products. It is both more durable and stronger than the aforementioned PLA, and can also maintain its shape in higher temperature situations. ABS can also be post processed with acetone to create a solid smooth surface with no layer separation marks. However, when 3D printing with ABS, during the print cycle as the plastic naturally cools, it is very prone to warping and disfiguring due to the contraction of the plastic. This means that it has a much higher failure rate when being printed.

When both of these materials were taken into consideration, ABS seems to be the more suitable plastic for the task. With the higher heat resistance, so less prone to deform if a servo over heats, and greater durability so that the system can withstand more continuous interactions, ABS wins in each regard. However due to previous experience working with ABS plastic, sometimes these beneficial factors are not enough to subtract away from its main flaw, warping. When a piece needs to be produced at a high level of accuracy, and many parts need to be rapidly prototyped, ABS becomes a very unreliable print material. For that reason, PLA will be used to develop the physical prototype of the system.

### 9.12.3 Flex 45

Flex 45 is a high carbon rubber filament that has been specifically developed for 3d printing. With a mixture of flexibility and a high mechanical strength, it is ideal for parts that need to absorb impacts or distribute loads. However, Flex 45 has no elastic property, a feature that many other rubber filaments have. This allows Flex 45 printed parts to deform completely depending on the environment they are in, but prevents their overall geometry from changing. As such a part made with this material can be placed under a high strain, without stretching which is ideal to produce soft robotic grippers.

### 9.12.4 Aluminium Extrusion

Key elements of the structure of the arm where modelled using finite element analysis to determine if extruded aluminium would be sufficient. The linear section is made of a 1" by 10swg aluminium box section with a 11mm thick, 10mm wide steel rack glued and bolted to the top. For the finite element analysis, this will be considered a perfect bond, as any deflection that results in delamination would cause unacceptable distortion.

The arm was first tested for deflection, by simulating a worst-case scenario, in terms of load. The arm is modelled as fully extended (700mm between tip and front support bearing) with a 20N load applied at the tip, the weight of the arm is also factored in (Fig 30).

This loading is more than the motors can hold, and as such is in excess of anything the arm is going to endure. The linear arm was found to have minimal deflection (27um), mostly due to the tensile strength of the rack. In practise the aluminium box section is far heavier duty than is needed.

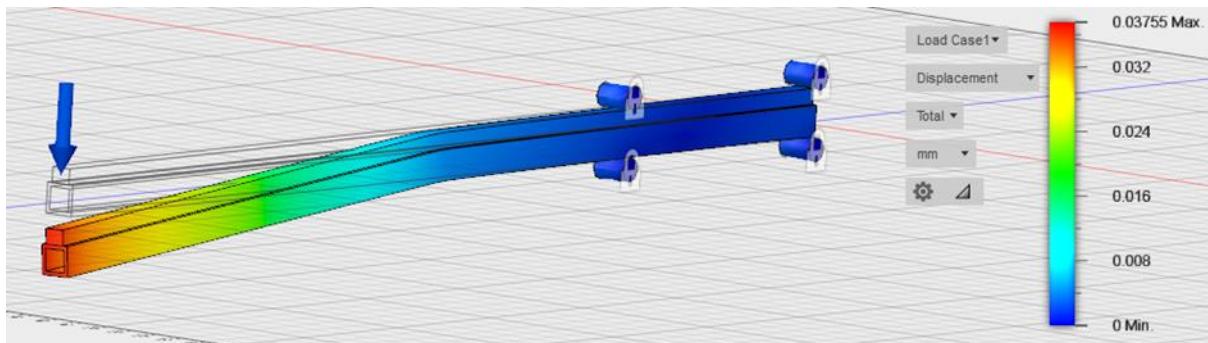


Fig. 30: FEA for deflection of the Prismatic section of the arm design.

The same scenario was also run to analyse stress (Fig. 31).

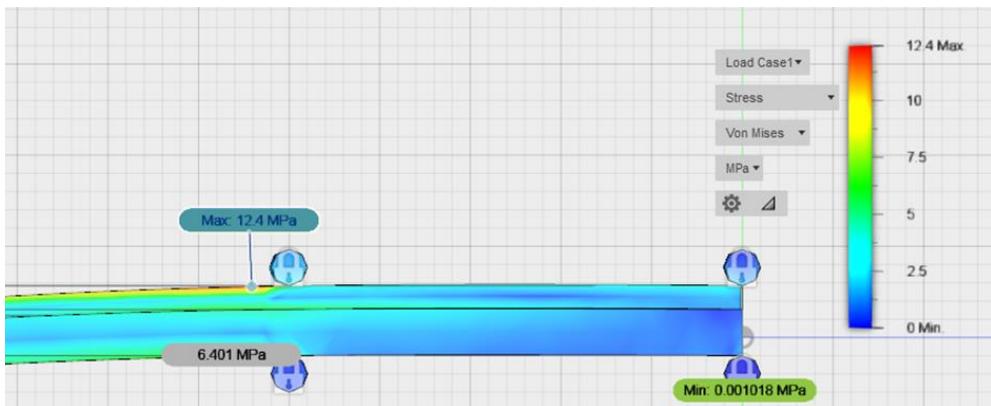


Fig. 31: FEA for stress of the Prismatic section of the arm design.

As the design puts the aluminium in compression and the steel in tension the arm is not at risk of exceeding the yield strength of either material (~250MPa for steel, and 75MPa for aluminium).

The second component that experiences any significant load, is the fork that support the pitching section (Fig. 32). The fork is made of the same 1'' square box as the linear arm, with a 4mm thick plate attached to the bottom. The entire structure is supported by a 12mm rod in the centre of the plate.

The fork is loaded with 50N per arm, to simulate the arms weight as well as a shock load from the arm trying to pitch rapidly. The fork is likely to experience oscillations as the arm rapidly pitches. As

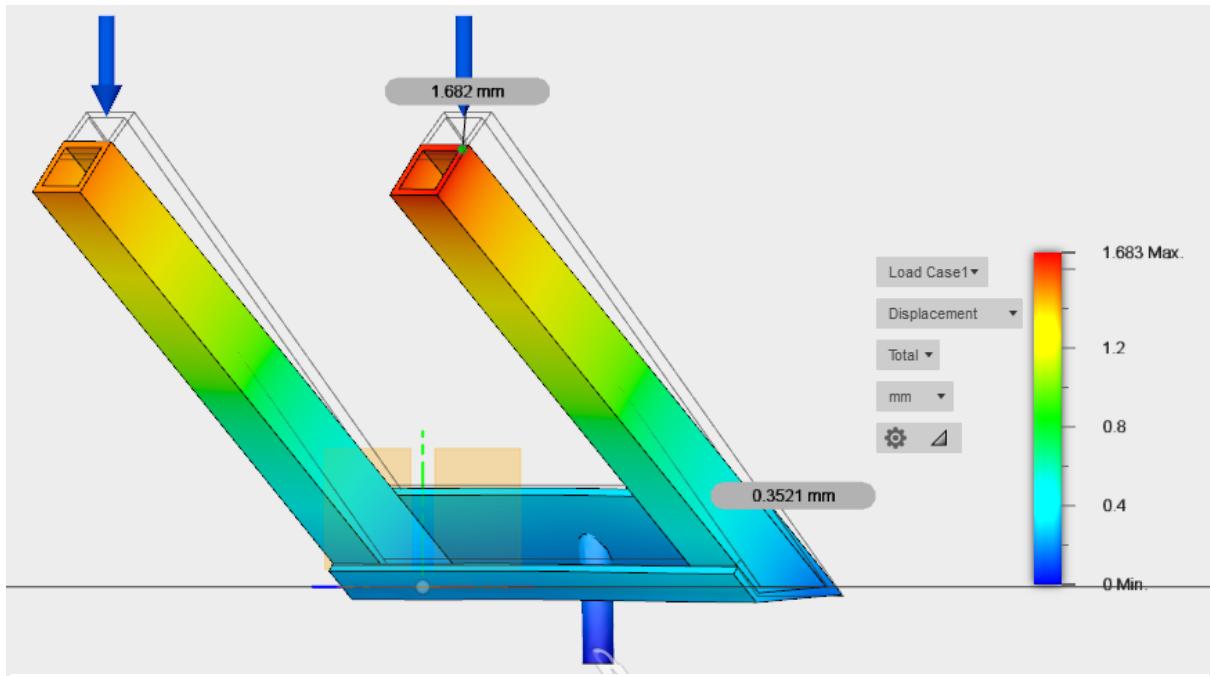


Fig. 32: FEA for deflection of the fork section of the arm design.

such the deflection given will be approximately half of the oscillation amplitude.

By bending the plate to support the base of each fork, the oscillation was brought down to an acceptable level  $\sim 3.2\text{mm}$ . Given that each fork can oscillate independently and thus in antiphase to each other, this is still greater than would be liked, but for a prototype, it is acceptable, and given that it uses the same material as the linear section, convenient.

## 10. Theory

This section outlines various theory factors that were implemented within the prototype.

### 10.1 I<sup>2</sup>C Communication

This project uses a two-wire interface, more commonly identified as I<sup>2</sup>C, to communicate between the Nano microcontroller and the servo driver board. This allows for easy expandability in the future as further sensors and modules can simply be added to the bus.

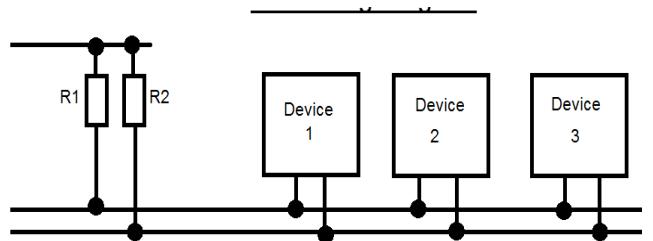
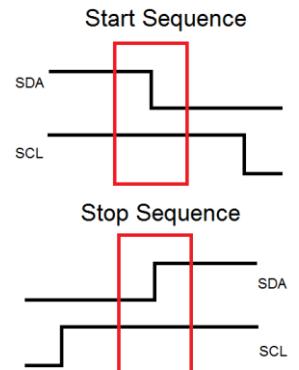


Fig. 33: I<sup>2</sup>C Wiring Diagram

I<sup>2</sup>C transmission is based around interfacing and sharing data between multiple devices by using only 2 lines. One line is the SCL and this is the clock line which is exclusively controlled by the master device. It is used to synchronise all the data transfers as the clock is pulsed every time a new bit is sent. The second line is called the SDA and that is the data line. Any device that wants to communicate over the I<sup>2</sup>C lines needs to be set as an open drain device, as both I<sup>2</sup>C lines are held high so the state can only change when the line is sunk to ground (Fig. 33).



To begin and end transmissions, start and stop sequences are sent respectively. This can be seen in Fig. 34.

Figure 34: I<sup>2</sup>C Start & Stop Sequences

From there data is transferred 8 bits at a time. Every time 8 bits are transmitted, the slave device then can “ping” an acknowledge bit to confirm it received the data. The first 8 bits it transmits are a 7-bit long address followed by a read or write command. From here the slave devices know if they are being talked to or not, if not, they do nothing until another start bit is detected. An overview of the transmission can be seen in Fig. 35.

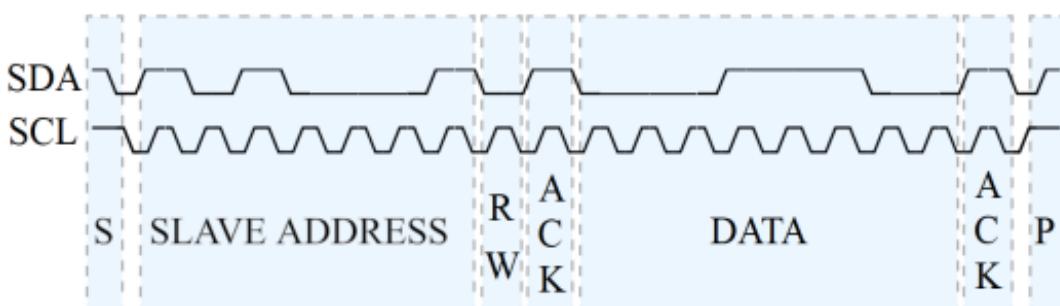


Fig. 35: An overview of the lines state changes taking place to perform successful I<sup>2</sup>C interactions [43]

## 10.2 Coordinate Frames

The origin for the robot arm was set at the yaw axis level with the bottom of the base. This was chosen because it allows for easier physical measurements when debugging and error checking. The X-axis is set to the front of the arm, the Y-axis is set to the left of the arm and the Z-axis is set to the vertical direction (Fig. 36).

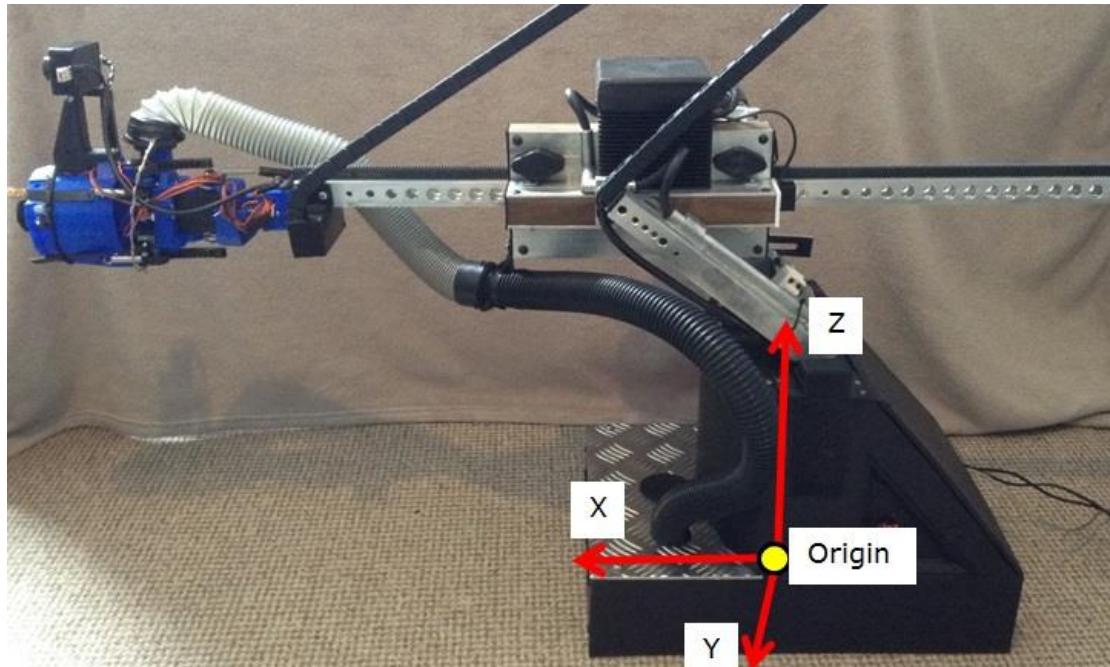


Fig 36: Base Coordinate Frame

Pitch is defined as the angle of intersection with the X-Y plane. A rotation upwards was defined as positive. Yaw is defined as the rotation about the Z-axis, with the X-axis direction being defined as  $0^\circ$ .

Several coordinate frames are used and related back to each other so that the position of strawberries in space can be calculated relative to the base of the robot arm. The main coordinate frame that the others relate back to is the base coordinate frame shown in figure 36. The other coordinate frames are the camera and end effector coordinate frames.

The camera coordinate frame has its origin set to the position of the left lens of the stereo camera, with the X-axis being in the direction the camera is pointing and the other axis orthogonal to it.

The end effector coordinate frame is in the same orientation to the camera's coordinate frame but with the origin offset to the center of the tip of the vacuum gripper.

## 10.3 Relating the Camera Coordinate Frame to the Base Coordinate Frame

First the camera coordinate frame is made to match the end effector coordinate frame by switching the axis and applying the camera offsets:

$$x_{coord} = z_{camera} + x_{cameraOffset}$$

$$y_{coord} = -x_{camera} + y_{cameraOffset}$$

$$z_{coord} = -y_{camera} + z_{cameraOffset}$$

Next the coordinate frame must be rotated to match the base coordinate frame. To remove the yaw component from the coordinates they are rotated about their Z-axis:

$$x_1 = x_{coord} \cdot \cos(-yaw) + y_{coord} \cdot \sin(-yaw)$$

$$y_1 = y_{coord} \cdot \cos(-yaw) - x_{coord} \cdot \sin(-yaw)$$

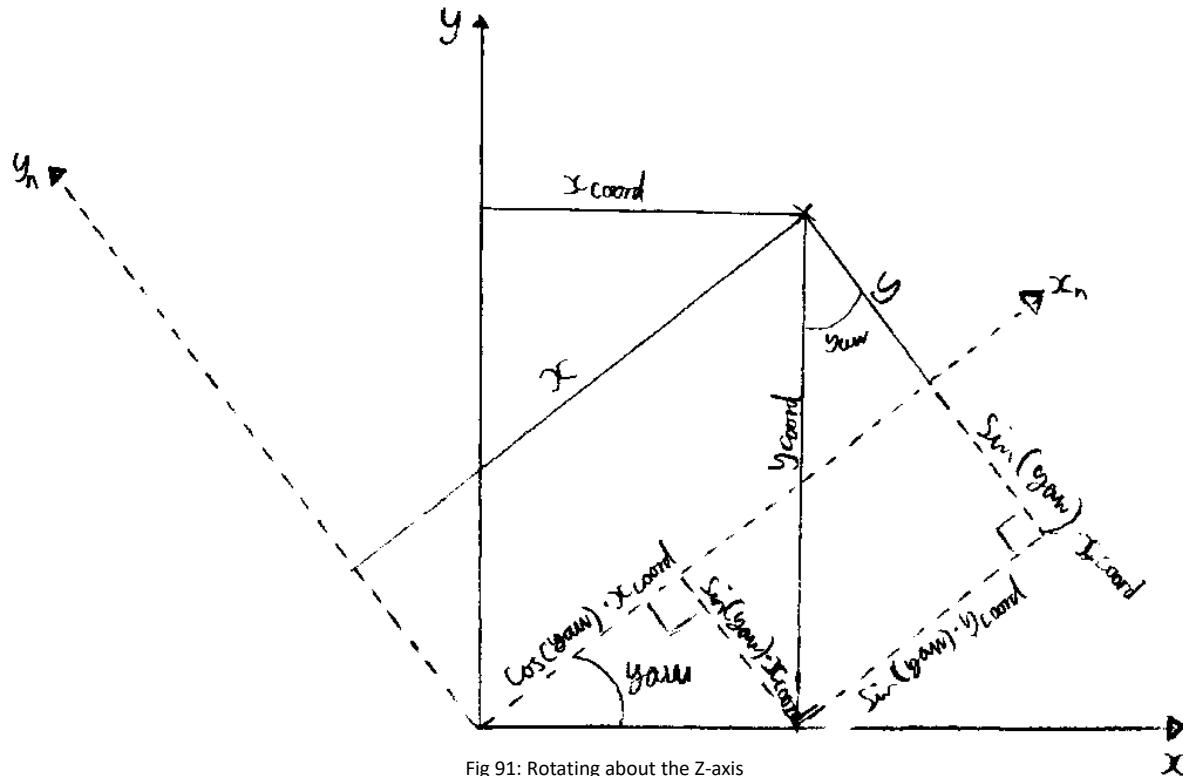


Fig 91: Rotating about the Z-axis

To remove the pitch component from the coordinates they are rotated about their Y-axis:

$$z_1 = z_{coord} \cdot \cos(-pitch) + x_{coord} \cdot \sin(pitch)$$

$$x_2 = x_{coord} \cdot \cos(-pitch) - z_{coord} \cdot \sin(-pitch)$$

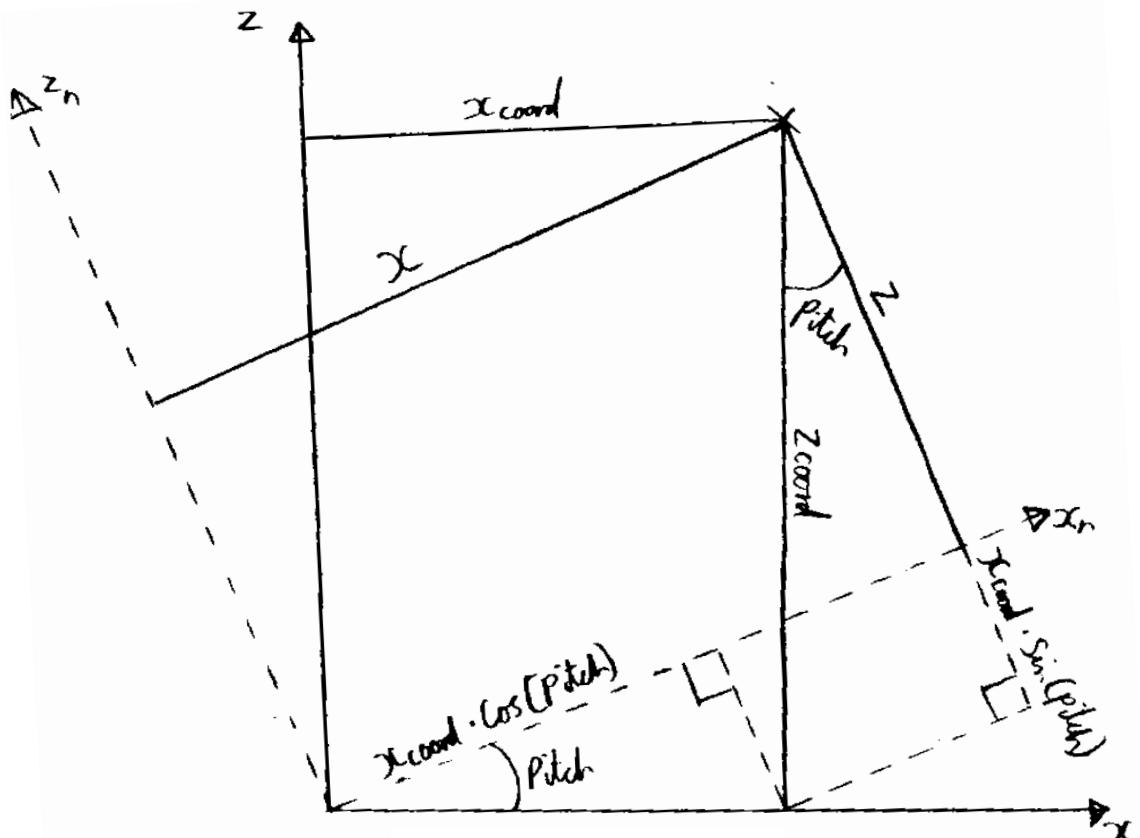


Fig 92: Rotating about the Y-axis

The current end effectors position must now be added to the coordinates so that they will be relative to the base:

$$x = x_2 + x_{endEffector}$$

$$y = y_1 + y_{endEffector}$$

$$z = z_1 + z_{endEffector}$$

## 10.4 Kinematics

Both forward and inverse kinematics was needed to give the desired positional control of the arm. The position of the arm is specified by giving the desired X, Y and Z coordinates for the end effector and giving the desired pitch and yaw of it.

The length of the end effector and orientation will create offsets that need to be subtracted from the target coordinates, to give the required coordinates at the rotational axis of the yaw servo, for calculating the joint positions. As the end effector pose is specified by the target position, forward kinematics can be used to resolve the required X, Y and Z offsets.

$$xy = x_{endEffector} \cdot \cos(pitch)$$

$$x_{offset} = xy \cdot \cos(yaw)$$

$$y_{offset} = xy \cdot \sin(yaw)$$

$$z_{offset} = x_{endEffector} \cdot \sin(pitch)$$

Due to the mechanical properties of the arm, the end effector drops in height proportionally to the extension of the linear segment. To counteract this drop and make the end effector reach the target coordinates, the term ' $z_{drop}$ ' is calculated and added to the Z coordinate.

$$x_{target} = x_{coord} - x_{offset}$$

$$y_{target} = y_{coord} - y_{offset}$$

$$z_{target} = z_{coord} - z_{offset}$$

$$z_{drop} = \frac{\text{linearExtension} - 183.4}{10}$$

Where linearExtension is the linear extension calculated using the same inverse kinematic equations with  $z_{drop}$  set to 0.

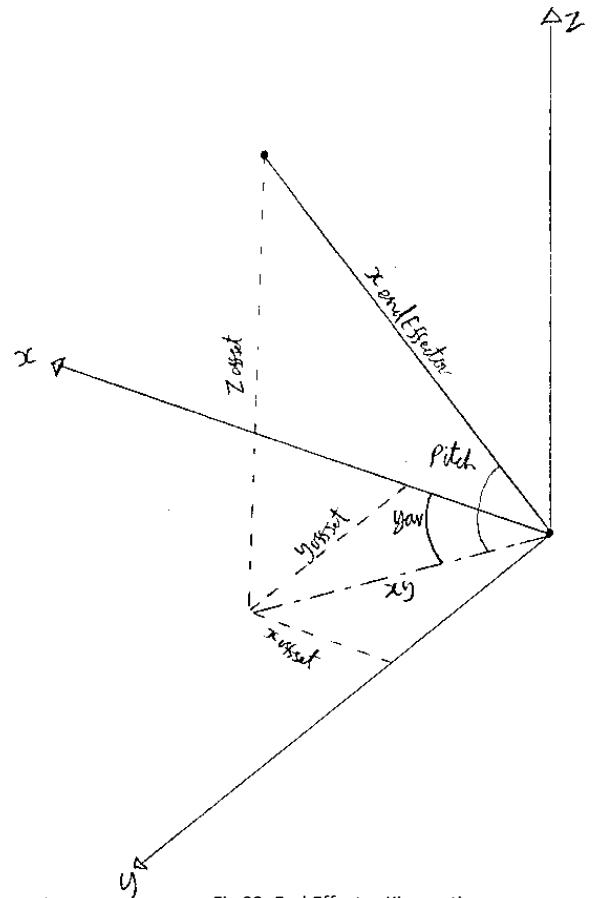


Fig 93: End Effector Kinematics

$$z_{diff} = z_{target} - z_{originOffset} + z_{drop}$$

$$yaw_{base} = \text{atan2}(y_{target}, x_{target})$$

$$xy_{diff} = \sqrt{x_{target}^2 + y_{target}^2}$$

$$xy_{pitchOffset} = link_{pitchToYaw} \cdot \cos(pitch)$$

$$z_{pitchOffset} = link_{pitchToYaw} \cdot \sin(pitch)$$

$$length = \sqrt{(xy_{diff} - xy_{pitchOffset})^2 + (z_{diff} - z_{pitchOffset})^2}$$

$$\text{linear} = \sqrt{length^2 - z_{linearOffset}^2} - linear_{offset}$$

$$pitch_{offset} = \sin^{-1} \left( \left| \frac{z_{linearOffset}}{length} \right| \right)$$

$$pitch_{base} = \text{atan2}(z_{diff}, xy_{diff}) + pitch_{offset}$$

$$pitch_{servo} = pitch - pitch_{base}$$

$$yaw_{servo} = (yaw - yaw_{base}) \cdot \cos(pitch)$$

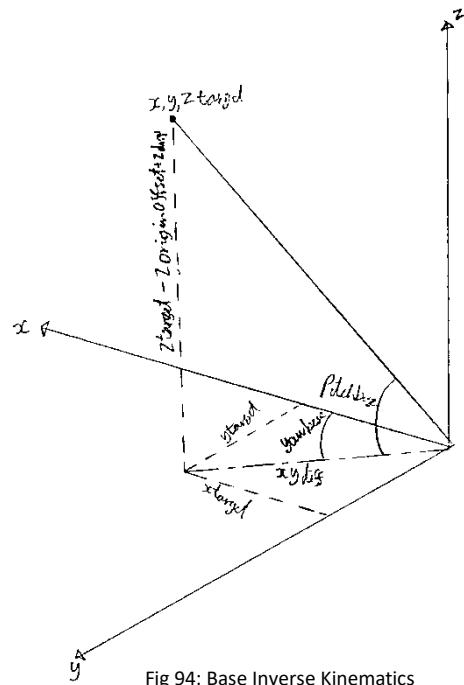


Fig 94: Base Inverse Kinematics

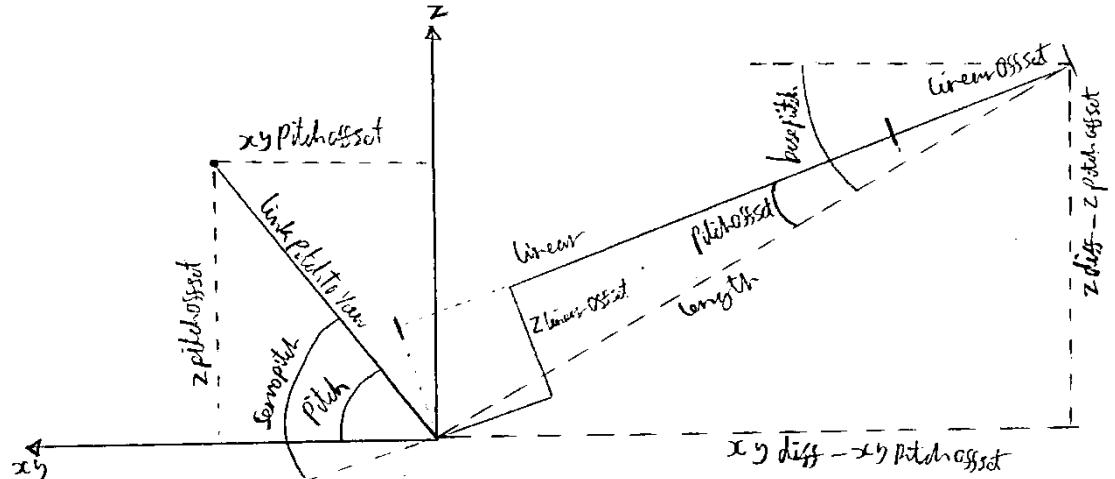


Fig 95: Linear Inverse Kinematics

## 10.5 PWM Servo Control

Servos are high torque position controlled motors with negative feedback to correct positional errors. They usually have a rotational limit between 0° to 180° however continuous rotation servos are also available but are not used in this project so will not be discussed further. The angle of the servo is controlled by sending it a PWM signal. The width of the high pulses determines the servos position. Pulses are normally sent every 20ms with the high pulse being between 1ms and 2ms with a 1ms pulse causing the servo to move to 0° and 2ms moving it to 180°. Figure 37 shows the timing diagram for this. The all servos are slightly different and may require slightly different length pulses to reach the target position so to increase the accuracy of the servos, the time of the pulses that will make it move to its minimum 0°position and its maximum 180° positon are noted and used to scale the pulses in-between allowing the servo to reach its desired position accurately.

Two different types of servo are used in the end effector, the Tower Pro 9g Micro servo SG90 and the HJ S3315D robot servo. Due to the servos being position controlled this mean that they move to then new target position as quickly as possible which in the case of the pitch and yaw servos was undesirable when trying to delicately handle strawberries. To prevent the end effector from suddenly jumping to a new position, small intermediate steps are used with a small delay between them.

The number of steps is the absolute value, of the largest change in angle in degrees, between the pitch and yaw.

$$pitch = pitch_{servo} + \frac{pitch_{difference} \cdot step_{count}}{steps}$$

Where  $pitch_{servo}$  is the pitch servo's position before the new target position was received,  $pitch_{difference}$  is the difference in the initial position of the pitch servo and its new target position,  $step_{count}$  is the current step number, and  $steps$  in the total number of steps the servo will take throughout its movement to the new position.

$$yaw = yaw_{servo} + \frac{yaw_{difference} \cdot step_{count}}{steps}$$

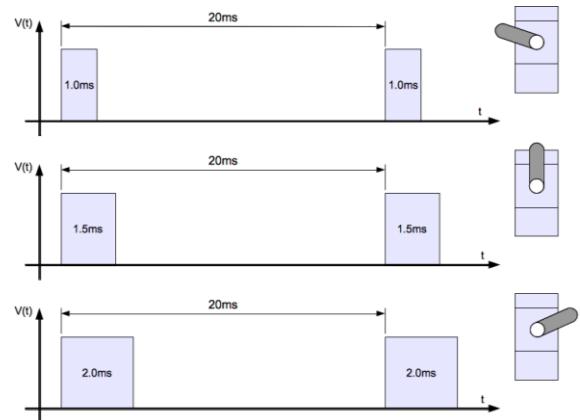


Fig 37: Servo Control [55]

Where  $yaw_{servo}$  is the yaw servo's position before the new target position was received,  $yaw_{difference}$  is the difference in the initial position of the yaw servo and its new target position.

Extending the sheath required the simultaneous movement of two micro servos in opposite directions. To ensure that they moved synchronously the calculations of the pulse counts of both servos was performed then the PWM signals were updated one after the other to minimize the time delay between.

## 10.6 DC Cutting Motor

The motors require a PWM frequency that is low enough to efficiently produce torque while being high enough that the coils inductance and the rotors inertia smooth the motors impulse responses caused by the PWM signal. The motors can be modelled as a RL circuit with the time constant being given by  $\tau = L/R$ . The voltage will rise to 63.2% after a single time constant and to 95% after three. Frequency for these time constants is then given by  $f = 1/\tau$  giving an appropriate range of PWM frequencies to drive the motors. The PWM driver board was set at 60Hz and as this was in the appropriate range of frequencies, it was used.

The motor need to be able to drive in both directions and as the PWM driver board's PWM channels are only signal line they will not supply sufficient current to power a motor, an H-bridge motor driver is required. The L9110 IC DIP [56] was used. Two of the PWM driver board's channels were used to control the motor driver by sending a PWM signal to the IA and IB pins. The motor driver was controlled via the PWM board because it was mounted physically near to the end effector and would not require extra wires being taken from the Arduino to the robot head.

## 10.7 Lens Distortion

Figure 38 shows the most common types of camera lens distortion. Any distortion needs to be removed from the images to allow stereo matching algorithms to work well. The cameras used has a slight barrel distortion thus when rectified the corners were stretched out and sides slightly curved inwards (figure 39).

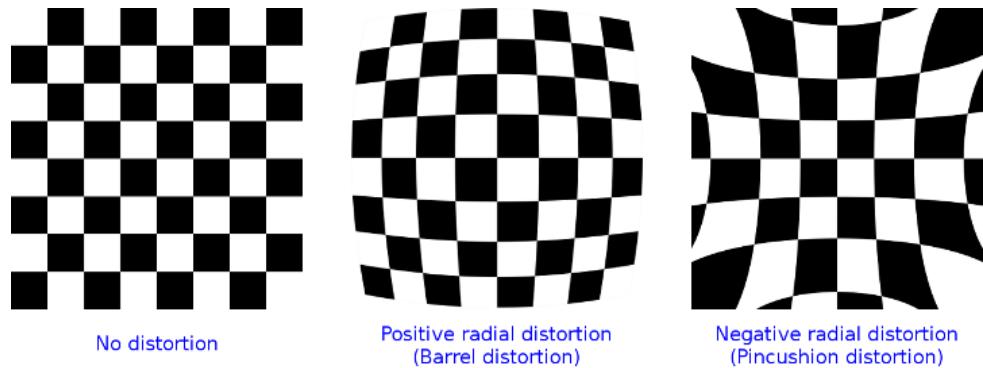


Fig 38: Lens Distortion

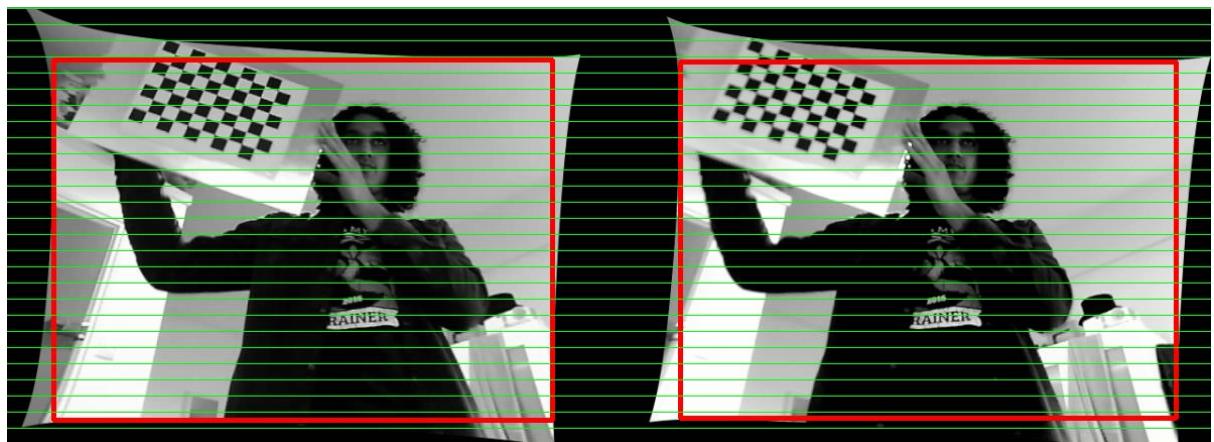


Fig 39: Lens Distortion Calibration [51]

To clearly display the disparity map to users, it was first normalized to a 0 to 255 range. The Laplacian edge algorithm was applied to the smoothed, grey scale, left image and the resulting image converted into a threshold image. It then a median blur was applied to reduce any noise before it was used to mask the normalized disparity. This resulted in clearly defined edges. Finally, the normalized disparity had the Jet colour map colour scheme (figure 40) applied to it. This makes closer objects appear redder and more distant objects appear blue. The colour map is not used for calculated only as a visual representation of the disparity map.



Fig 40: Jet Colour Scheme

## 10.8 Calculating Coordinates Relative to the Camera

The coordinates relative to the camera use the Z-axis as distance from the camera (depth), the X-axis as horizontal displacement from the optical axis and the Y-axis as the vertical displacement from the optical axis with down being the positive direction.

$$z = \frac{\text{baseline} \cdot \text{focalLength}}{\text{disparity}} \cdot 0.873$$

$$x = \frac{(px - cx) \cdot z}{\text{focalLength}} \cdot 1.471$$

$$y = \frac{(py - cy) \cdot z}{\text{focalLength}} \cdot 1.471$$

Where 0.873 and 1.471 are scalars to give the correct values in mm, px is the X-pixel coordinate, cx is the center X-pixel coordinate, py is the Y-pixel coordinate and cy is the center Y-pixel coordinate.

## 10.9 Controlling the Relay

The Arduino Nano GPOIs are only rated to supply 20mA. The relay requires a higher current to operate therefore another NPN transistor circuit was required. The relay was switched using 5V instead of 6V however the circuit design was kept the same for simplicities sake (figure 41).

## 10.10 Limit Switches

The arm has four limit switches (Fig. 42) which it used for the homing operations and as a hardware mechanism to prevent movements which could damage the arm or signify an effort in the code. The switches are all connected to each other and are in a pulldown configurations. This mean is any of the switches are pressed then the connected Arduino will be pulled down to ground.

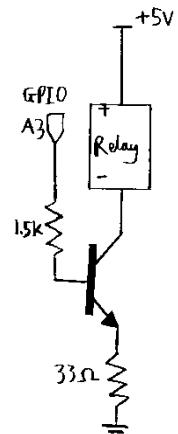


Fig 41: Relay NPN Circuit



Fig 42: Limit Switch

After every pulse is sent to the stepper motors the limit switch state is checked to ensure that the arm has not hit any of them. If a switch is pressed then the code will print an error message and stop.

## 10.11 Simultaneous Stepper Motor Movements

All the stepper motors are moved to their new target position simultaneously to avoid making each motor wait for the previous one to finish before moving, and to make the movement between points more predictable.

To make all the motor move together and finish the movements at the same time, first the target angles are converted to pulse counts, then they have current position counts taken from them to give the change in pulse counts. With the known change, the direction the steppers will need to move in can be determined and set. The number of steps taken to complete the movement is set to be equal to the largest number of pulses that any of the motor will take. The count spacing between each motor's pulses is then given by the largest number of pulses divided by that motor number of pulses. A motor is made to step if the step number is larger or equal to that motors count spacing times the number of pulses that motor has already done.

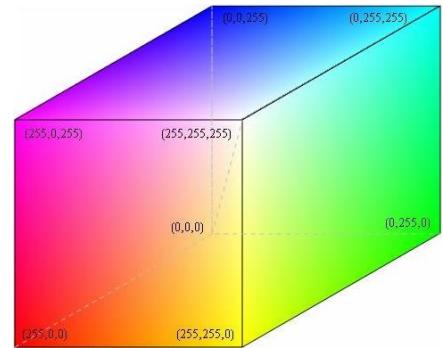


Fig 43: RGB Colour Space [47]

The red-green-blue (RGB) colour space is a cartesian cube representation of colour (Fig. 43) that has red, green and blue as its three primary colours that all others are formed from. The default colour space of images being processed by the OpenCV libraries is BGR, which is the same as the RGB format but with the bits storing the red and blue channels information swapped. The bit depth of each RGB channel depends on the datatype used to store it however 8-bit and 16-bit are frequently used for this application. They give each channel a range of 0-255 and 0-65535 respectively.

The hue saturation value (HSV) colour space is a cylindrical coordinate representation of colour (Fig. 44). OpenCV can convert from the RGB colour space to the HSV colour space which gives some advantages when trying to filter for colours of a certain hue. Hue, saturation and value are all stored in 8-bits each. This would allow a maximum value of 255. Hue normally has a range of 0-360° however this could not be stored in a byte so OpenCV uses hue with the full range scaled to 0-180°. As HSV is a cylindrical colour space that starts with 0° being pure red, this means that the hue wraps back around to 0° after its maximum value, this give a red hue at both the minimum and maximum ends of the spectrum as can be seen in figure 45.

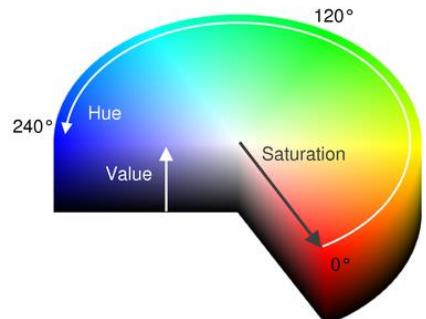


Fig 44: HSV Colour Space Cylindrical [48]



Fig 45: HSV Colour Space Linear [49]

## 10.13 Path Planning

When the target is identified its coordinates and the pose of the end effector are stored in an array. If a target can be seen there must therefore be a clear line of sight to it although it may not be large enough of the end effector to move through.

With the target position and end effector pose the approach vector can be calculated. The approach vector will take the form of the straight line formed by connecting the targets coordinates and end effector's pose from the target positions array.

The end effector cross-sectional shape is known so collisions with any objects can be avoided by checking this area about the approach vector. The disparity map often contains a small amount of noise so any small objects are assumed to be errors and if any would intersect with the end effectors path they are simply ignored.

When the end effector has moved along the approach vector and is 260mm from the target, the stereo camera is used to take three stereo image pairs and the mode average filtering algorithm is applied to them. This then gives the most accurate updated coordinate for the target allowing the end effector to precisely move to it.

Once a strawberry has been picked the end effector is retracted back along the approach vector ensuring that it does not collide with any obstacles. The strawberry is then placed into the storage container.

## 10.14 Object Identification

The aim of the project is to pick ripe strawberries therefore they will need to be a sufficiently red colour. This allows the recognition algorithm to primarily be based on colour.

Working in the OpenCV HSV colour space the hue boundary for a ripe strawberry was set to be  $\pm 15$ , while the saturation and value were variable dependent upon lighting conditions. As previously mentioned a red hue is at 0 and wraps around at 180 so a boundary of  $\pm 15$  required two threshold operations one for the range of 0-15 and another for the wrapped around range of 165-180. Both threshold images then need to be combined to give the all the matching pixels. Figure 46 shows the detected colours on a printed hue wheel, the left most image is with a hue range of 0-15, the middle image is with a range of 165-180 and the right most image is them combined. (Due to printer and camera characteristics colours may not be accurate but are useful for visualization and debugging.)

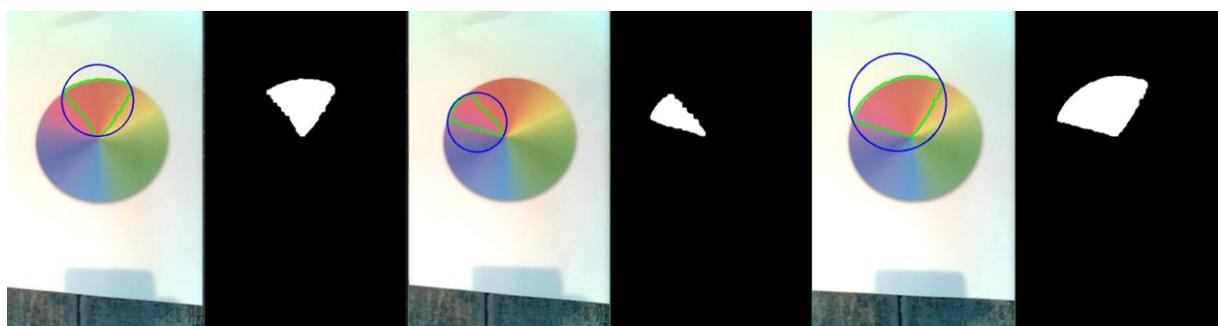


Fig 46: Hue Thresholds

In an attempt to make the recognition as robust as possible, the RGB colour space was also used to create a threshold image that was combined with the other two. Figure 47 shows the separated RGB channels of the right most image of the hue wheel (the channels are displayed in their respective colours). The brighter that the colour of the channel the more of that colour component was in the original image, this is most notable with the white paper as it appears equally bright in each channel. This was expected as white in the RGB colour space is made by combining the three colours. This means a red strawberry cannot simply be detected by looking at the red channel alone as this would show any colours that had a red component.

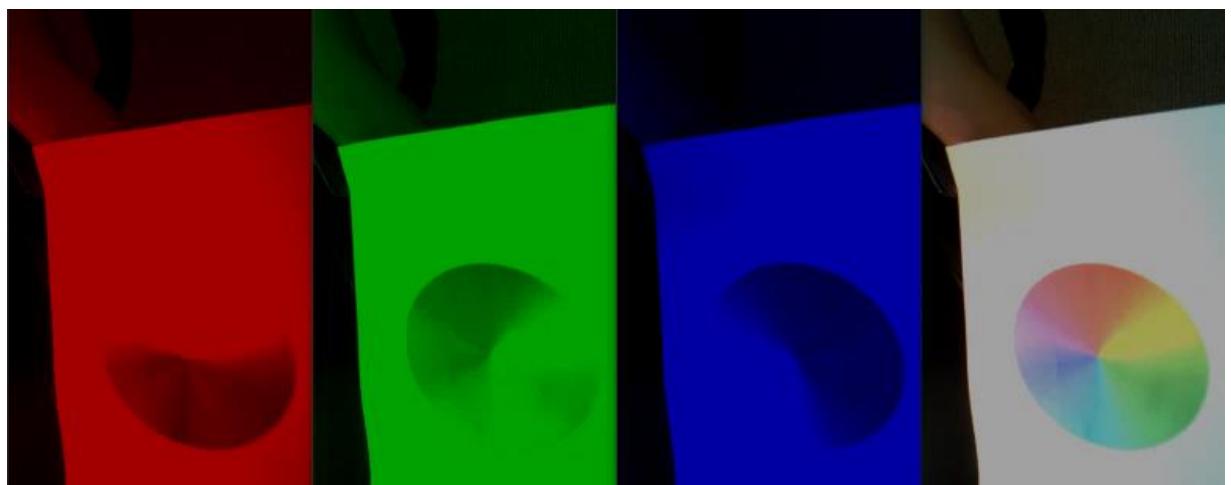


Fig 47: RGB Channels

To extract just the red objects in an RGB image first the image was split into its separate channels, then the blue and green channels were subtracted from the red channel. This only left objects that were close to pure red contained in the red channel. The green and blue channels then had their pixels set to a value of zero, before being recombined with the red channel. A threshold was then applied to this recombined image. The threshold value had

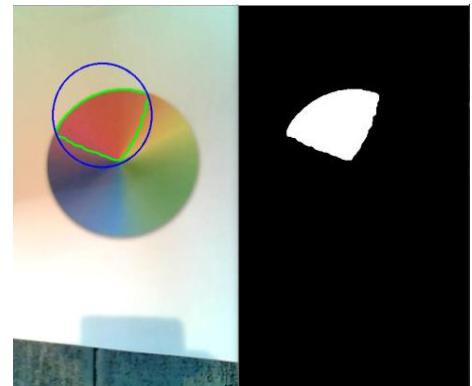


Fig 48: Red Channel Threshold  
a dynamic minimum value to help deal with changing lighting conditions and the maximum was set to 255 (the maximum value an 8-bit channel can hold). Figure 48 shows the identified target colour and associated threshold image.

The ripe strawberries have a large value in the red channel and much lower value in the green and blue channel thus appearing darker (figure 49).



Fig 49: Strawberry RGB Channels

The threshold image created from the RGB channels is added to the threshold image created from the hue range to give a more reliable composite threshold image. This composite image has noise reduction techniques applied to it to remove speckles and fill holes.

The erode function is used to remove small speckle. After the erosion has been applied, a median blur is used to further remove any remaining speckles and to smooth the edges of the target areas. The dilate function is used fill any holes but due to the erosion the holes would have grown in size, thus requiring the dilation to have 5 iterations. This causes the target area to be larger than the original size so requires another application of the erode function.

## 10.15 Ripeness Estimation

There are three stages that the ripeness is checked. The first is initial detections of the strawberries, they must be sufficiently ripe to be in the correct colour range to be detected and they must have an area larger than the threshold. The second stage is far more accurate and happens when the stereo camera is moved just in front of the target strawberry to reassess the target coordinates. The final check is performed when the strawberry is held in the vacuum gripper. This requires the cameras to be tilted downwards by 45° (Fig. 50) so the target is in both cameras field of view.

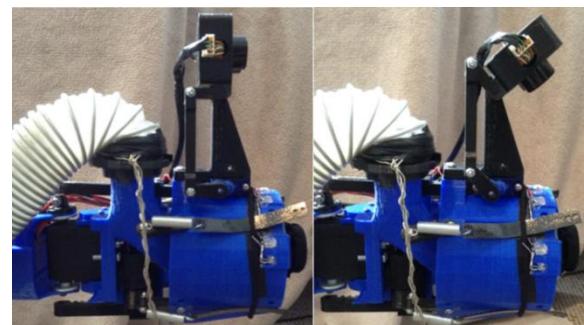


Fig 50: Camera Tilt

When a detected strawberry is selected to be picked the cameras on the end effector are moved to 260mm away from it while pointing in its direction. A distance of 260mm was chosen because it is short enough that the inaccuracies remain small and far enough that an accurate

disparity map can be generated, even if the initial coordinates were inaccurate by several centimeters. A delay was added to allow any movement in the arm to stop before three new pairs of stereo images are taken and the position of the target strawberry extracted. The mode averaging algorithm is then applied to the coordinates obtained from these frames to determine the most accurate position of the target. With the distance known, distance and known size of the target area the physical size of the strawberry can be calculated. If the matched area of the strawberry is too small then it is considered unripe. However, if the matched area is larger than the threshold for a single strawberry, it is considered to be a bunch of strawberries that the Laplacian edge detection had not correctly broken apart. In this case, the closest part of the matched area is selected as the target.

The end effector is then moved to the target which is then gripped via the vacuum gripper. When the target strawberry has been gripped the end effector is retracted slightly to ensure that there are no leaves obscuring it, it is separate from any bunches and that the vacuum gripper has a sufficient hold of it.

After the end effector has moved back, the gripper sheath is retracted if it is not already, the ring LEDs switched on, the camera tilted downwards. The sheath should always be retracted except when a stem is about to be cut, however it is better to have redundancy built in. If the sheath is extended then a gripped strawberry will not be obscured by it. The ring LEDs provide a diffused structured lighting environment for the cameras allowing more reliable and consistent colour detection.

The camera is in a fixed location relative to the gripper nozzle. Therefore, a gripped strawberry will appear in the same location every time for each camera. This allows the image processing to be simplified by masking and removing the excess area around the nozzle (figure 51).

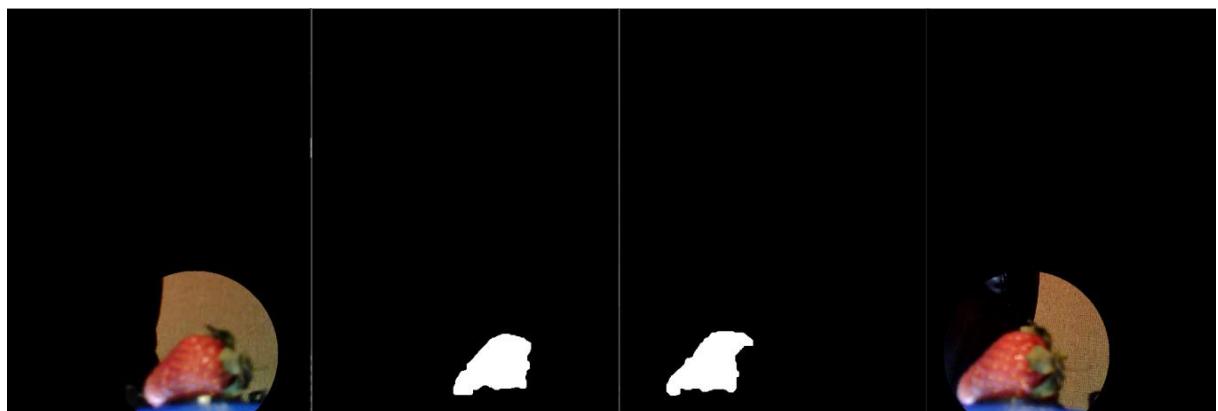


Fig 51: Ripeness Estimation

Both cameras are used to filter for the colour as the both have slightly different viewing angles so produce two different images. The average of the two target areas is taken so it can be compared to the estimated surface area of the held strawberry.

One of the biggest difficulties is that the strawberry may be held in an orientation that puts the stem and calyx facing the cameras. This will prevent most of the ripe parts of the strawberry from being seen thus making a ripeness estimation more difficult.

To estimate the ripe surface area of the strawberry an estimate of the size needs to be made. The strawberry is held at a set distance from the camera therefore the disparity from a stereo image pair will always be approximately the same. This allows the disparity map to be filtered and a threshold image created. The area of the threshold image is taken as the area of the strawberry and used to determine the percentage area of the strawberry that is ripe.

The threshold of the strawberry area is used to make a binary mask that is used on the left camera image. This leaves just the strawberry, calyx and stem in the image. The calyx and stem have a deep green hue whereas the unripe parts of a strawberry are a much paler whitish green so can be differentiated. If only red and the calyx/stem green are detected then the strawberry is assumed to be ripe. Although this is not a perfect solution it gives satisfactory results due to the fact that the last place to ripen is normally around the stem of the strawberry therefore if the calyx and stem are in view the unripe parts should be too.

$$Area_{avg} = \frac{Area_{left} + Area_{right}}{2}$$

$$Ripeness_{percentage} = \frac{Area_{avg}}{Disparity_{area}} \cdot 100$$

Note that this method of estimating the percentage ripeness would allow the percentage to be over 100% if there are inaccuracies in determining the target areas and size of the strawberry. Only a lower threshold is used to choose if the strawberry will be picked therefore it does not matter if the percentage ripeness is over 100%. If the ripeness percentage is too high it may be an indication that there was an error in detecting the size thus requiring the estimation algorithm to be run again.

## 10.16 Strawberry Identification

The identification algorithm was initially developed using a red ball as a strawberry analogue. This was done because strawberries have a short shelf life once picked so a new supply would be needed every few days and this was not practical for a serval month long project. Using a strawberry analogue gave the advantage of being able to compare algorithms without having a different target as added variable.

Once the algorithms were developed actual strawberries and strawberry plants were used for testing. Six punnets of strawberries were purchased from different stores, to give the most varied range of strawberries, and combined they contained 126. They were randomly mixed and used for testing the algorithms.

Figure 52 and figure 53 show batches of 42 strawberries being used to test the identification on a flat surface. This allowed the HSV saturation and value thresholds to be calibrated.

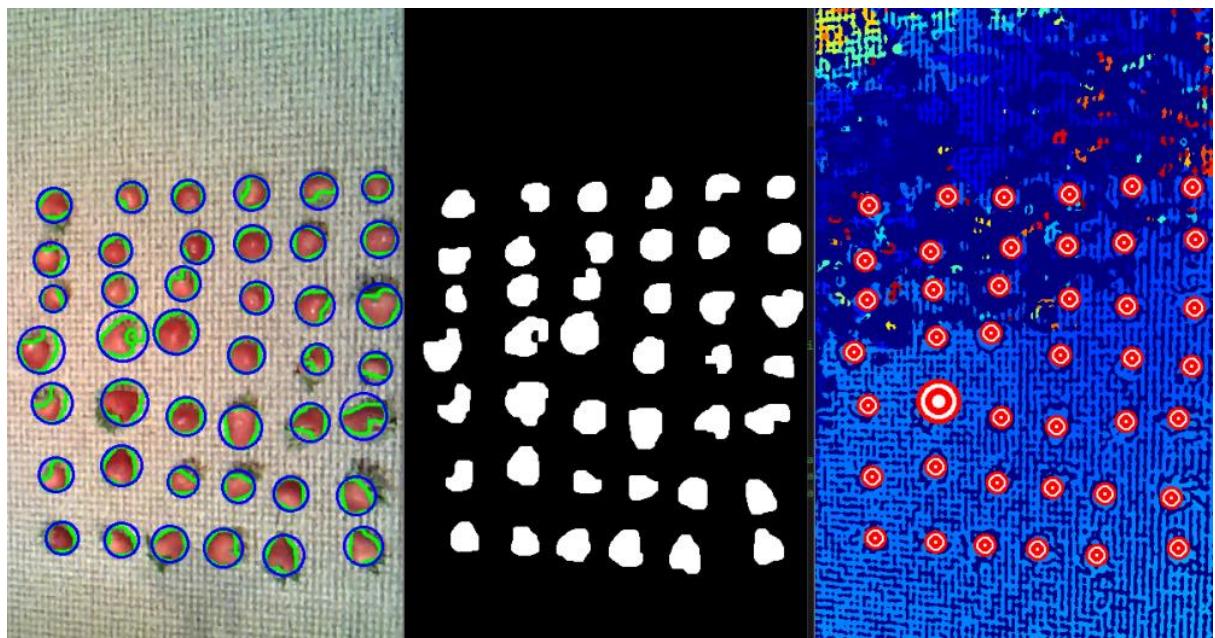


Fig 52: Strawberry Saturation/Value Calibration 1

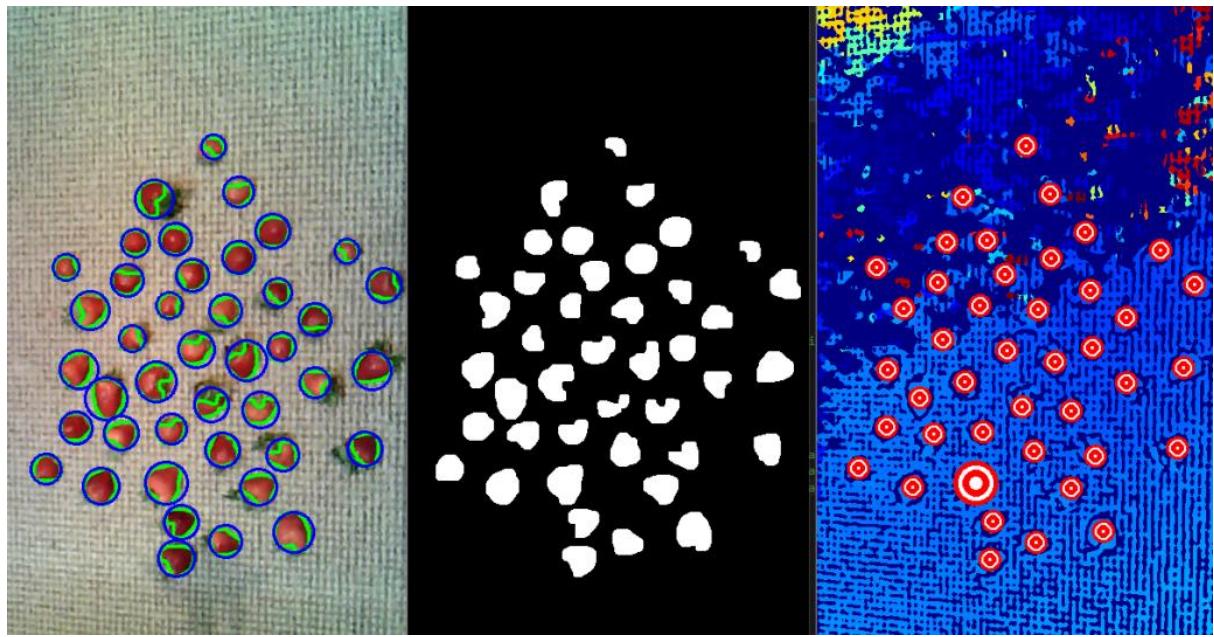


Fig 53: Strawberry Saturation/Value Calibration 2

Detection on the bought strawberry plants was tested. The plants had a few unripe strawberries growing on them therefore some of the ripe strawberries that had been bought were placed on the plants. Figure 54 and figure 55 show the strawberry detection on the plants. Various distances and lighting conditions were tested to determine the limits of the systems.

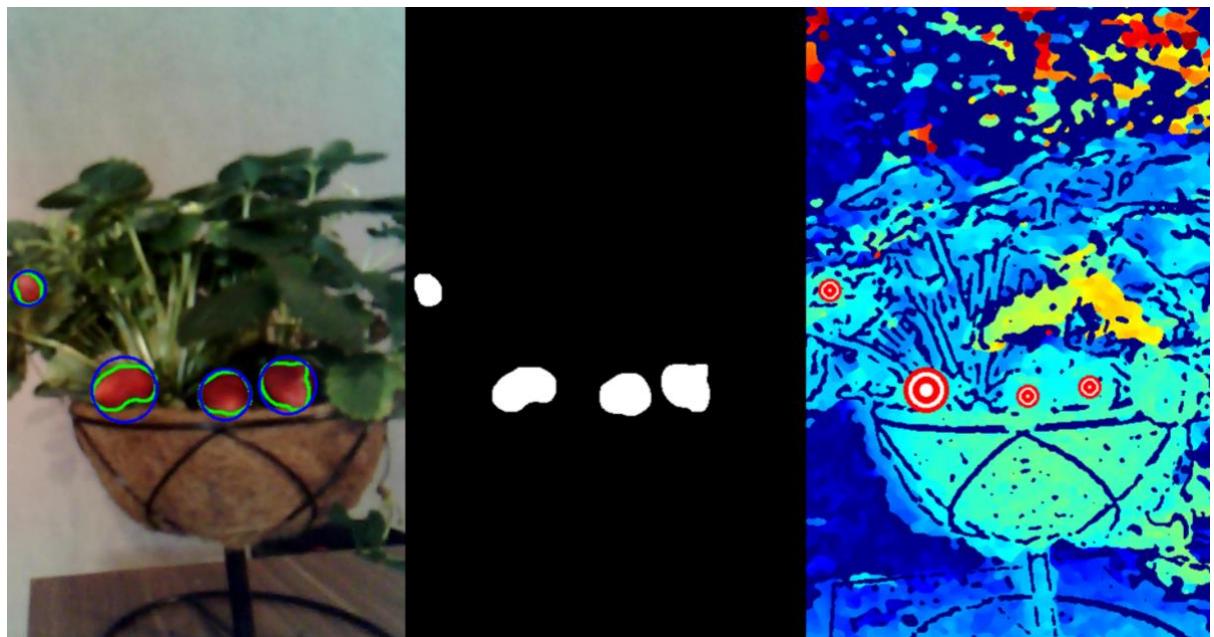


Fig 54: Strawberry Plant Detection 1

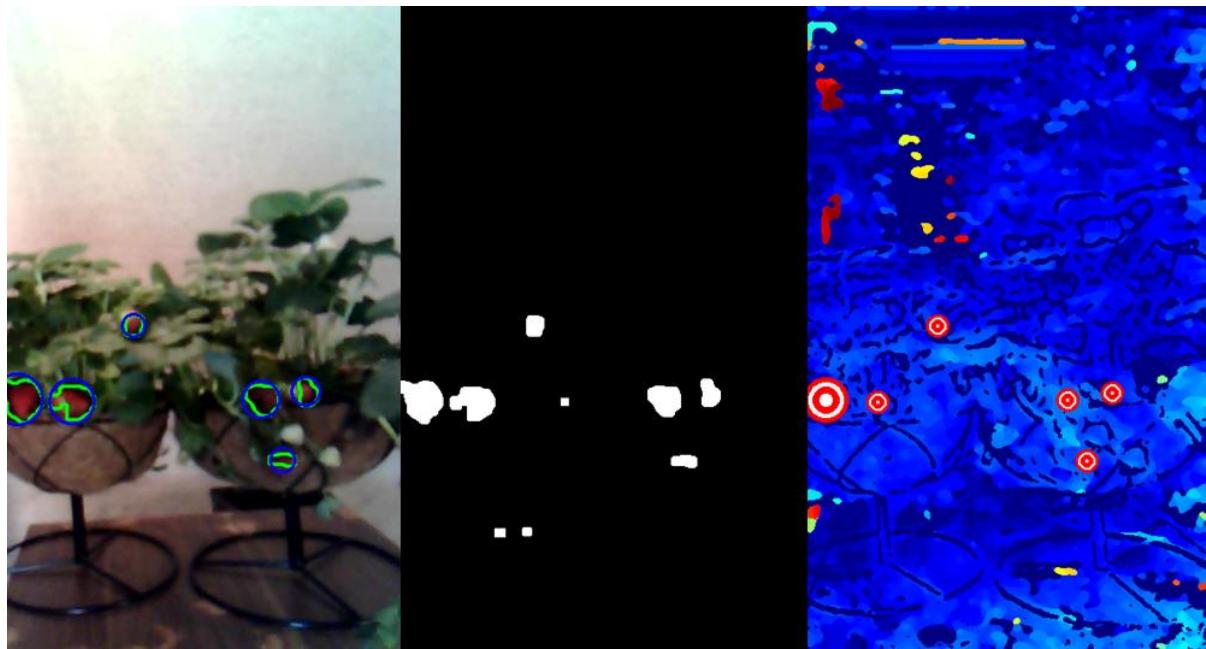


Fig 55: Strawberry Plant Detection 2

In good lighting conditions strawberries were reliably identified however detection rates drastically dropped when there was not enough light or in too bright a light.

The results of dim lighting were expected as the cameras were not able to detect colour correctly in such conditions. The dim light would cause the automatic white balance to adjust and cause matching on some of the dark brown objects.

The unforeseen issue with bright lighting was that the cameras picked up a bright white glare from the brightly lit objects. This glare obscures the target red colours preventing a match being detected.

It is possible that both of these issues could be fixed by using better quality cameras, that deal with more varied lighting conditions better and that do not have automatic white balance. The cameras used have a CMOS sensor thus the blooming effect from bright lights is very significant. Using cameras that have a CCD sensor would reduce the blooming and potentially allow for correct colour matching.

## 10.17 Algorithm Detection Testing

To test the reliability of the identification algorithm in good lighting conditions (a room with a main light and two lamps), six strawberries were spread over two strawberry plants in changing configurations. For comparison, a human viewed the camera stream and identified all the strawberries they could see in it. This was the best the algorithm could reasonably be expected to achieve with the camera input. Figure 56 shows the results.

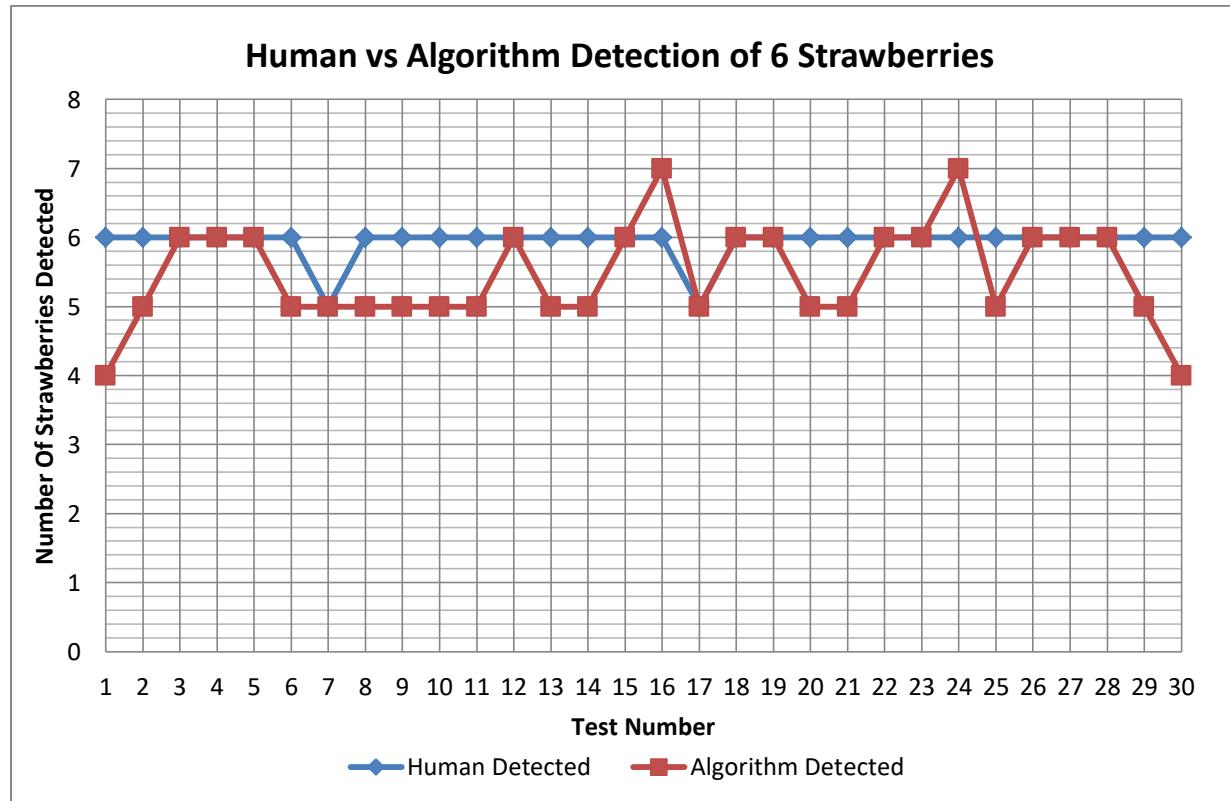


Fig 56: Human vs Algorithm Graph

Total number of strawberries in the tests: 180

Human errors: 2

Algorithm Errors: 18

$$\text{Human: } \frac{180 - 2}{180} \cdot 100 = 98.89\% \text{ Identification Rate}$$

$$\text{Algorithm: } \frac{180 - 18}{180} \cdot 100 = 90\% \text{ Identification Rate}$$

The most notable causes for the algorithm failing to detect a strawberry, was having a said strawberry in deep shadows cast by the leaves.

## 10.18 Stereo Distance Estimation

It would be feasible or necessary to test the distance estimation of the stereo cameras at every possible distance. The distance estimation was tested at 600mm from the camera, 600mm from the camera with the mode averaging algorithm applied and 260mm. The distance was selected because the distance from the cameras to edge of the arms workspace straight ahead was 600mm when it was in its home position during development. The distance that the cameras are moved in front of the target position before picking for position revaluation is 260mm.

Due to the characteristic function of stereo cameras measurement errors increase with distance thus it can be assumed the a distance of 260mm will have the smallest errors and 600mm will have the largest. Distances between these values should therefore have errors between the error values at these two distances.

To test the precision and accuracy of the stereo camera a target was set 600mm from the camera and the distance estimation made. This was repeated 30 times and the results recorded in figure 57. Keeping the same distance to the target the distance was then estimated 30 times with the mode averaging algorithm applied to give the results shown in figure 58. The target was moved 260mm from the stereo camera and distance estimated implementing the mode averaging algorithm, the results are shown in figure 59.

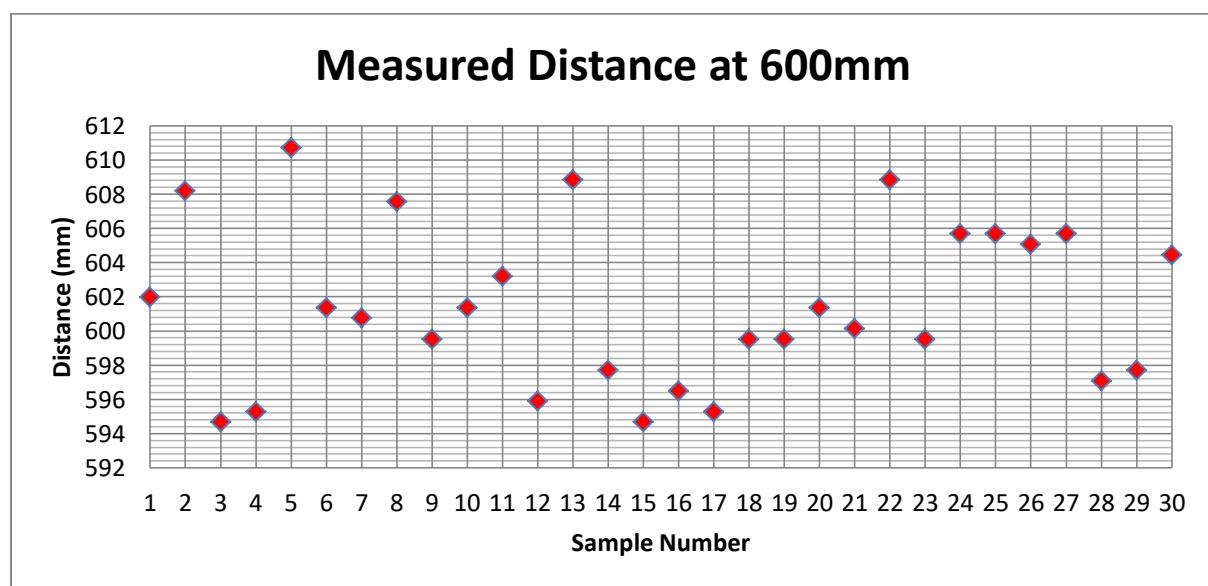


Fig 57: Distance Estimation 600mm

## Mode Average Measured distance at 600mm

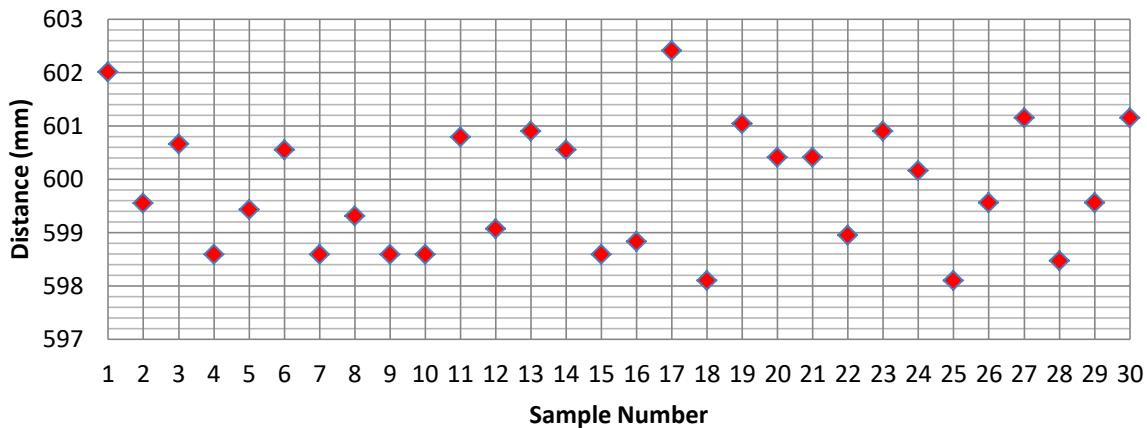


Fig 58: Distance Estimation with mode Averaging 600mm

## Mode Average Measured Distance at 260mm

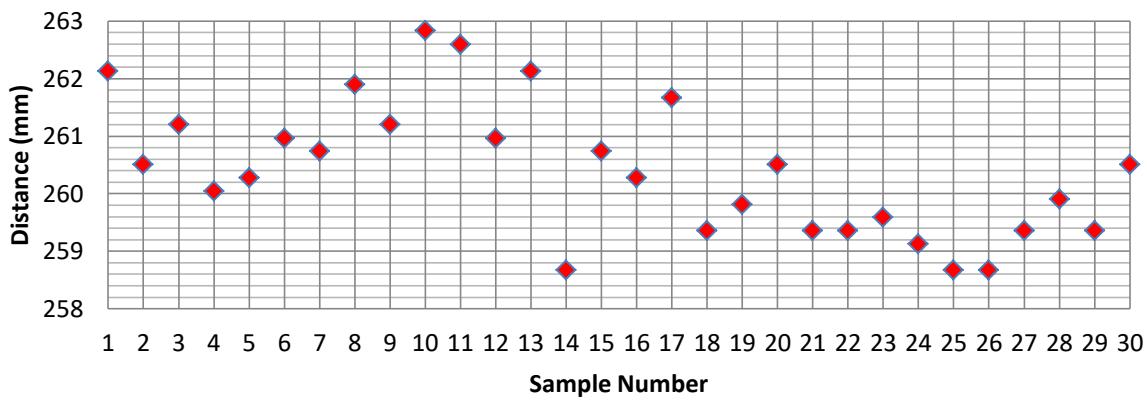


Fig 59: Distance Estimation 260mm

The average value and standard deviation of each data set is show in table 3. As the distances were set to 600mm and 260mm respectively these were the expected averages. As can be seen in the table there is a slight discrepancy however it is very small and there are two likely explanations for it. The first is that there are only 30 measurements so the average could be easily skewed by a few results. Secondly the initial set distance may have been measured incorrectly. With an error this small no action needed to be taken to rectify it.

	Average	Standard Deviation
Measured at 600mm	601.4628333	4.635418716
Mode Average Measured at 600mm	599.8330333	1.164122545
Mode Average Measured at 260mm	260.4104667	1.156319787

Table 3: Average and Standard

## **II. Prototype design**

This section outlines aspects regarding the physical design and build of the project. This includes design choices and justification.

### **II.1 Arm**

When considering designs for the arm, three main ideas were suggested.

The first was to use a traditional robot arm, i.e. a series of pivoting joints, much like a human arm. This gives the advantage of a lot of dexterity and a good-sized workspace.

The issue with this is that an arm is complex to control without resorting to prebuilt kinematic solvers, such as ROS [44]. Traditional robot arms also need high quality motors and encoders for accurate control, which, given a minimum of five motors would be needed, would likely exceed the available budget.

The second suggestion was to use a Gantry design with a wrist like end-effector that sits over the bush. Gantry arms are quick and easy to control and can be built using inexpensive motors.

The issue with a gantry design is that it will always be trying to access the fruit from above. given that strawberries are almost always covered by foliage and thus cannot be accessed from the top, a complex wrist would be needed to access the strawberries from the side.

The concern with this design was that the wrist would be too bulky and would damage the plant as it picked fruit.

The final suggestion was to use an arm with a linear section as its main source of movement. Such a design would use two high power, high precision, motors to pitch and yaw an extending linear section and cheaper, low power servos could be used to articulate a wrist.

This means that only two motors are moving a significant load, unlike all five for a traditional arm, meaning that the design can fit within the available budget.

This design also has the ability to ‘look and grab’, a process where the pitch and yaw can be lined up with a target, and the linear section extended to grab it. This simple control method can be made to not require a kinematic model to function, working instead on mechanical processing, potential making control very easy if desired.

As such, for the mechanical design of the arm a three-axis, revolute-revolute-prismatic design was decided upon (Fig. 60). This was chosen as it concentrates the electrical system and the moving mass, close to the base of the arm, as well as reducing the cross-section of the arm.

By concentrating the weight around the base, it allows for higher end-effector velocity's, in theory increasing the speed that the arm can pick fruit, as well as reducing the strain on the motors caused by the arms momentum.

By concentrating the electrical components for the arm at the base, it maintains a distance between the electronics and the hostile environment of the field and allows the aluminium structure of the arm to be used to dissipate the heat generated. Because of this the entire electronic system for the arm can be passively cooled, removing the need for fans or vents, which in turn improves the systems resilience to its hostile environment.

For the revolute joints a direct drive linear stepper motor was used in order to minimise moving parts, again to improve resilience. Due to the torque required to manipulate an end-effector at the intended range of the prismatic joint, a planetary gearbox and a secondary chain drive was used, to provide sufficient torque with standard size motors.

For the prismatic joint a floating box, driven by a rack and pinion was chosen (Fig. 61) as it gives a large range of movements, and is resilient to fouling. It also gives a convenient conduit to route cables and pipes through, minimising the cross-section of the arm and providing protection for the cabling.

The use of rollers to support the rack, allows for a reduction in weight in comparison to linear bearing systems, both in terms of gross weight and in moving weight. Rollers are also cheaper, more resilient to fouling and rigid enough for the light duty work that the arm will be engaged in.

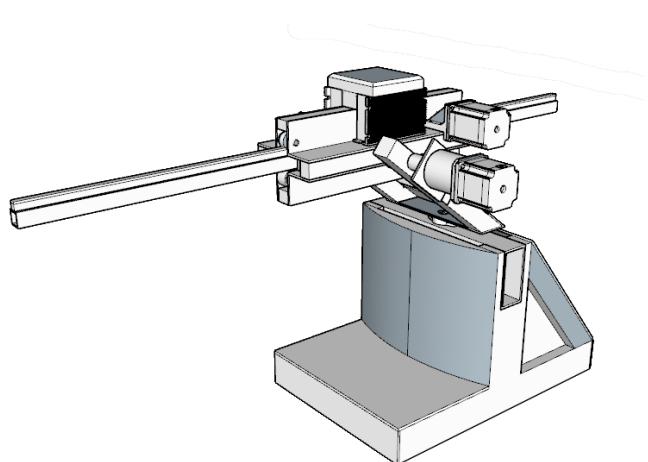


Fig. 60: The Final Arm Design

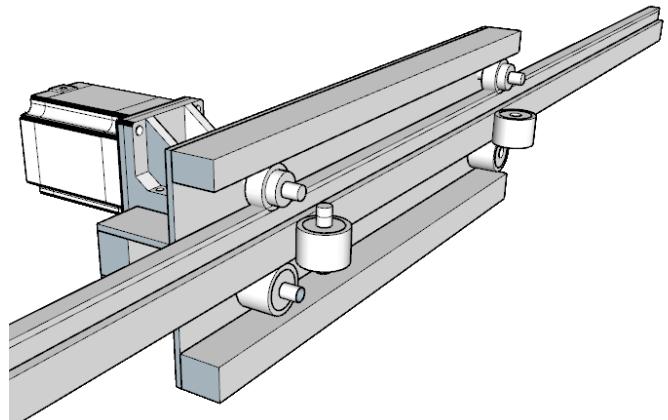


Fig. 61: The Arm Prismatic Joint Design

## 11.2 End Effector

The design of the end effector of this unit is a critical feature as it can determine the level of complexity required elsewhere within the machine vision of the project. For example, if the end effector requires precise positioning to grasp the stem of the fruit, advance visionary tricks would be required to both identify the thin stem, and then position the gripper appropriately around it. However, if the end effector can grasp and manipulate the fruit from any orientation upon its surface, then the visionary guidance can be far simpler and less precise.

Ideally, the gripping mechanism needs to be as physically small as possible to allow for both, insertion below the vegetation of the plant, and manipulation of the fruits when they are in close proximity to one another.

### 11.2.1 Grasping methods

This section examines various possible mechanical solutions for end effector gripper designs, and compares the advantages and disadvantages with one another.

### 11.2.2 Solid gripper designs

The simplest of possible gripper designs, and by far the most common, is a set of solid mechanical grippers. The design of such requires no complex material properties, instead opting for a fundamental limited movement.

#### 11.2.2.1 Clamp grasp

The first design considered and created for the gripping mechanism follows the concept of simplicity. With 2 solid fingers linked and actuated by a single servo motor (Fig. 62), this design would require stem identification. Once the stem has been recognised, the gripper would close in and pinch the stem. The soft silicon pads on either side would provide a soft grip so as not to crush or drop the stem, whilst the blade at the top of the finger would end flush with the solid surface of the finger, thus severing the stem. From here, the strawberry can be manipulated away from the plant and safely deposited.

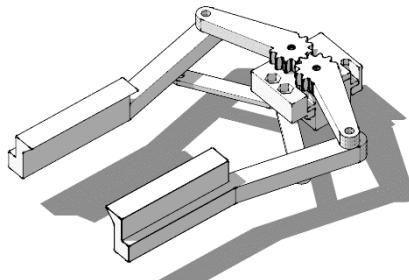


Fig. 62: Clamp Grasp

### **11.2.2.2 Enclosure Design**

The solid enclosing design developed consists of a scoop and a mechanical iris around its brim driven by a small dc motor. The idea is that the fruit is positioned into the scoop, with then the iris closing in, cutting the stem free. This would require excess room to be made available around the strawberry for the scoop to enclose it, and for the stem to also be identified so that the iris can be slid over the fruit.

### **11.2.2.3 Advantages and Disadvantages**

#### **Clamp grasp:**

This provides a very simple multi-action solution that performs the task of both holding the strawberry for transportation, and cutting it free from the stem. The overall size of the design can retain a small footprint, instead opting for very long cutting fingers to reduce the risk of damaging any of the fruits surroundings. Lastly, this design is unaffected by the size of the grown fruit as their stems always remain within a much smaller tolerance.

However, this design relies heavily on the prospect of analysing the fruit and determining their orientation, then their attached stem. This requires a very high resolution depth map image and direct object recognition of a strawberry, which is a very complicated challenge for the machine vision. As such, this design may not be the most appropriate one to use, and instead a design that can simplify the entire visionary guidance processes would be preferable.

#### **Enclosure:**

The advantage of this design is the fact that during this whole process, the strawberry is contained within a soft, safe enclosure, preventing the possibility of harm being caused due to environmental circumstance. However, this design would require the ability to get under the fruit, a feature that is not always possible with fruit that are growing upon the ground. It would only effectively work on individual produce that are separately growing.

### **11.2.3 Deformable gripper designs**

By using more complex material property combinations, it is possible to create designs that have dynamic stiffness attributes. This in turn allows for the creation of grippers that can deform to the shape of the object that they are grasping without the application of excessive force to their surface.

### II.2.3.1 Deforming Fingers

By introducing perpendicular support lines from the intended contact surface of the finger, and connecting these to a secondary adjacent wall, a complex deformation structure can be generated. When pressure is applied to the inside surface of the finger, the rigid perpendicular supports prevent the contact surface from collapsing. Instead the exterior wall is forced outwards and deform

with respect to its fixed pivots. As a result, the surface in contact with the fruit deforms and naturally curls the end of the two exterior walls inwards. By modifying the angle that these connecting perpendicular supports are drawn in, the deformation properties of the part can be altered to suit the needs desired.

This design uses this unique sub structure to create a pair of grasping fingers that deform to fit the shape of the strawberry (Fig. 63 and Fig. 64). Not only this, but as the end of the fingers retract in, due to the de-formation, the strawberry is centred and locked inside the now enclosed fingers.



Fig. 63: Deforming Fingers

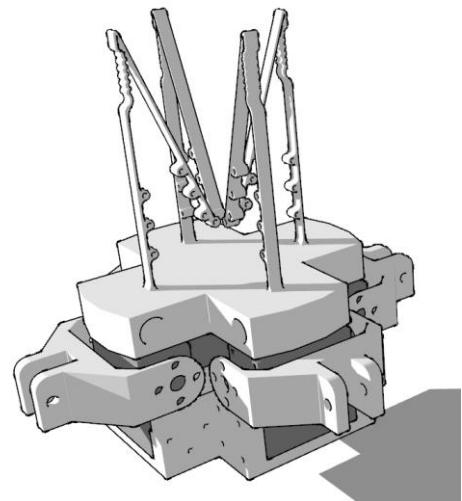


Fig. 64: Deforming Fingers

### II.2.3.2 Stem Cutting

To cut the stem from the enclosed strawberry, requires the use of a small secondary arm with a camera, and a cutting mechanism. It can use this pre-focussed camera to orbit around the fruit, not only for ripeness evaluation, but to determine the stems contact location. Once discovered, the small cutting mechanism is then able to move in and sever the stem free.

### II.2.3.3 Advantages and Disadvantages

The soft deformable fingers allow for the strawberry to be manipulated in almost any orientation, whilst providing it with a soft enclosure to keep it safe. However, upon further testing, some notable difficulties do arise. If a strawberry is growing in a cluster for example, the gripper must push past the first fruit, then when it begins to close, there is a chance that one of the

remaining fruits gets caught. This can result in either damaging the other fruit, or causing it to fall from the plant. In addition, the complexity of the cutting mechanism and stem identifying structure adds even more size to the gripper, resulting in a large end-effector that is not suitable for positioning under foliage of the plantations.

#### 11.2.4 Vacuum testing

Another method of manipulation that was tested was by using a vacuum. If a gentle vacuum can be created upon the head of the end effector, then the strawberry can be picked up and manipulated from any exterior surface. It would not matter its current orientation.

As such, a multitude of tests were performed upon strawberries with various intensity vacuum pressures using many different contact surfaces. After each test, the strawberry was then left for 48 hours at room temperature to see if any surface damages or bruising occurred. The results of which then shaped how the final vacuum nozzle was designed.

##### 11.2.4.1 Contact Surfaces

During testing, 4 base contact surfaces were tested to see which performed the most desirably.

Surface 1: The first surface tested was a rigid cylindrical nozzle. This provided no surface cushioning to the strawberry and was used as a control. Once the pressure inside the chamber dropped to a low enough value, for the strawberry to remain in contact during manipulation, the strawberry suffered serious surface damage.

Surface 2: The second surface was a soft fabric nozzle (Fig 65), supported by craft foam inside. This design was also able to hold the strawberry, however required the most powerful vacuum to do so. This is due to the breathable material letting air escape from around the fruit. The strawberry did not suffer any surface damage.



Fig. 65: Fabric Nozzle

Surface 3: A conical design allowing for the strawberry to fit inside the outer perimeter. This design proved to be the most reliable to allow for constant adhesion across a broad range of vacuum pressures. In addition, no damage was caused to the fruit.

Surface 4: This was similar to the first tested contact surface, however, was a soft cylindrical chamber. When the strawberry came into contact, the chamber reduced in size as the pressure behind the contact region fell. This design required the least amount of power to maintain adhesion to the fruit, however was still susceptible to damaging the surface.

From these it was determined that the best route to take would be to design a nozzle that maintains a conical and soft contact region with the fruit, whilst having the facility to retract into a soft neck section (Fig 66). This deforming capability ensured that, once contact had been made with the target, it would reshape itself to then match with the contours of said object. This required a variable wall thickness of the part so that certain elements would compress before others. This was to prevent the nozzle from retracting inside out, or from deforming away from the desired fruits contours during the low-pressure application.

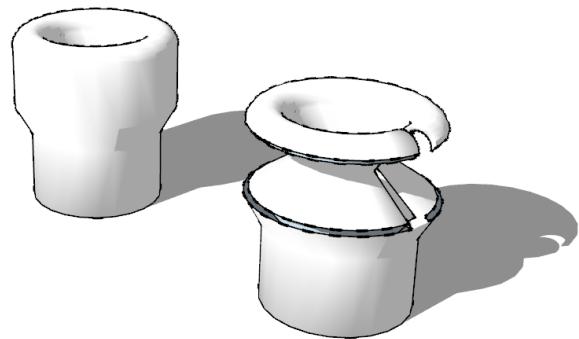


Fig. 66: Soft Conical Nozzle

### II.2.4.2 Fan design

As the vacuum tests proved, manipulating strawberries without damaging them, with a vacuum, is feasible. For that reason, research began on the construction of an appropriate fan design that could be used to generate the vacuum, on-board the unit. One that is capable of producing a suitable static pressure to hold the picked fruit.

Turbine fans de-pressure air as it draws it through and as such, experimentation began by rapid prototyping a turbine fan. The design was based from basic theory and other existing fan designs. This was then connected to a generic 12V DC Motor and enclosed inside of a container to act as housing. The air flow produced showed promising results so a refined housing (Fig. 67) was developed to see if the generated pressure could be increased.

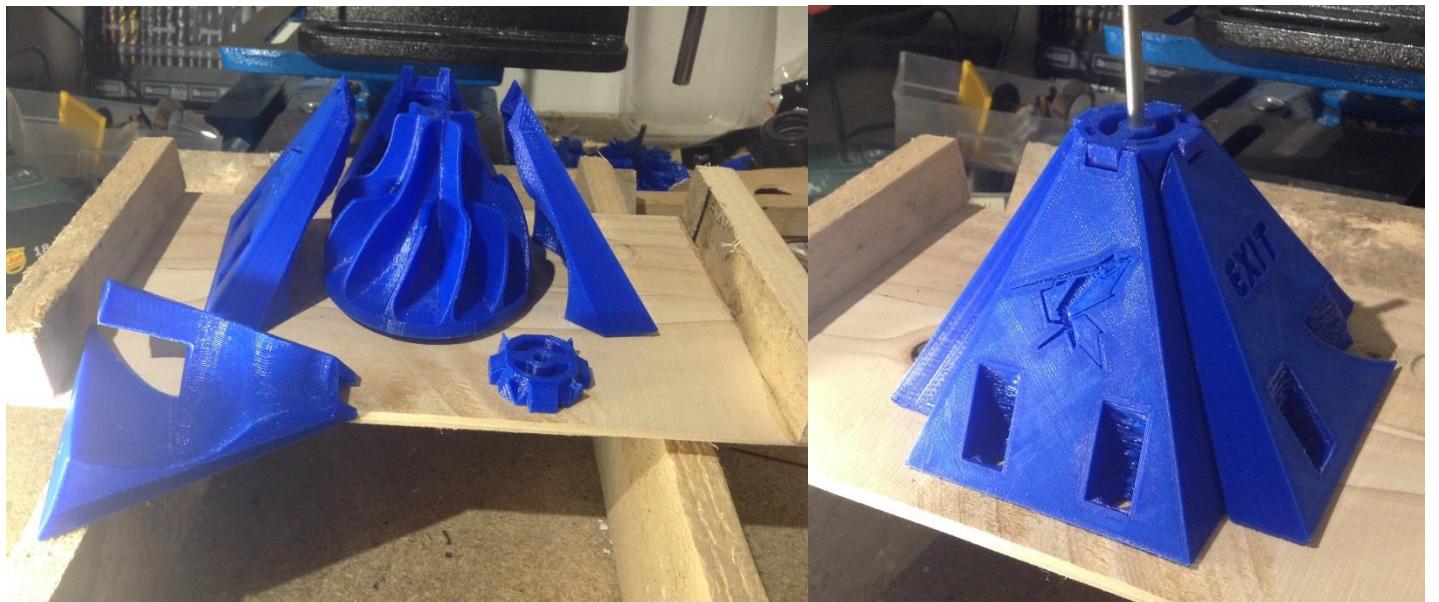


Fig. 67: Fan Housing

This developed design was capable of producing a static pressure; however it was not enough to match the requirements desired. Due to time constraints, rather than developing this design further, a commercial centrifugal YDK Fan assembly [45] was implemented instead. This runs from mains voltage and could produce a much greater air flow and pressure. As this prototype was intended solely for demonstration purposes under a short time span, it was decided that using a mains powered device would be acceptable. If this project were to be carried forward, such an aspect would need to be further developed and redesigned to suit a battery powered unit.

To control this motor unit, an opto-isolated solid-state relay was used [46], allowing for PWM control. The relay includes a zero-crossing point detection, this allows a switching rate of 50Hz and prevents high load switching.

## 11.2.5 Combining elements

The final gripper design (Fig. 69) borrows aspects from each of the previous iterations, and builds upon their concepts to create a final unit that can achieve the desired task. By using the vacuum contact technique for fruit manipulation, this design then has two rack and pinion drives (Fig. 68) which extend the cylindrical head forwards, encompassing the strawberry.

The advantage of this method is that a strawberry can then be grasped, from any orientation; the arm can then retract, pulling the strawberry away from other surrounding fruits. Once away, the head is then able to extend over, containing the strawberry inside of a safe enclosed space.



Fig. 68: Rack and Pinion

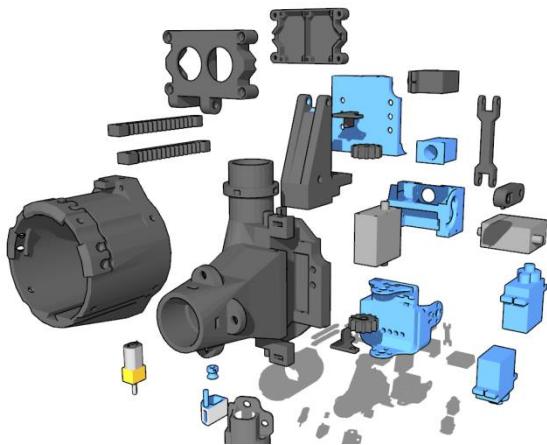


Fig. 69: Exploded CAD View

To cut the stem of the strawberry, rather than using a complex stem finding algorithm, a simple ‘soft’ solution was used. This uses a geared DC motor to wind in a nylon line draw string. This draw string encloses around the head of the unit, forcing the stem to be orientated outwards from the gripper, and it continues to close until the stem is forcefully cut by the pressure applied to the cutting sled. Once cut, the motor can then be back driven and the springs cause the nylon to stretch back out, ready for

another fruit. This acts as both a cutting mechanism for them stem, and a safe enclosure mechanism for the fruit as it further aids to hold the fruit inside of the safe head section.

The body of the unit has a mounting plate at its rear for easy connection to the wrist manipulator. On top of the unit is a flexible locking joint for the vacuum hose to connect to. This connects through a channel that travels through the whole body to the nozzle slot. This slot allows for nozzles to be quickly swapped between for testing purposes.

## 11.2.6 Structured Lighting

As this system needs to be able to perform visual analysis upon the surface details of strawberries, there needs to be some form of consistency between photographed elements. This requires structured lighting to ensure an even comparison can be made.

In this scenario, the camera needs to look at the fruit, and clearly detect the colour gradient so that the ripeness can be evaluated. To achieve this, a lighting set up must be in place which does not produce any harsh shadows upon the produce. In addition, the lighting used needs to be a strong white source, so as not to skew the resulting images. The best structure for this is a ring light.

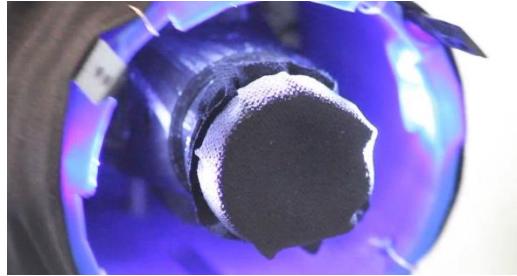


Fig. 70: Ring LEDs

With an evenly diffused ring of ultrabright white LEDs, the subject is brightly illuminated and no shadows are cast across it.

The LEDs need to be PWM controlled so that their brightness can be varied to give the best lighting conditions. If the camera frame rate is not a factor of the PWM frequency of the LEDs then there may be a flickering effect visible on the camera feed. As the LEDs are only used for ripeness checking which uses a stereo image pair not a continuous camera feed, they will not be interfering in this way.

The PWM signal lines of the driver board cannot supply enough current to power all the ring LEDs therefore a NPN 2N3904 [57] transistor circuit was needed to control the power to them (figure 71).

Eight ultrabright white LEDs [58] are used to form the structured lighting ring. The LEDs have a maximum current rating of 30mA and the NPN transistor has a current rating of 200mA.

A 200mA current limit would allow each LED in parallel to have:

$$\frac{200mA}{8} = 25mA$$

Therefore, this is a perfectly acceptable transistor to use.

The minimum value resistor to give a 200mA limit is therefore:

$$\frac{6V}{200mA} = 30\Omega$$

A 33Ω resistor was used as this keeps the current below the limit and allows the LEDs to sufficient power.

$$\frac{6V}{33\Omega} = 181.818mA$$

$$\frac{181.818mA}{8} = 22.727mA \text{ per LED}$$

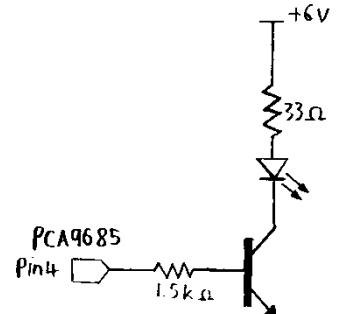


Fig 71: LED NPN Circuit

### 11.2.7 Camera positioning

The stereo cameras need to be able to both, scan the environment to identify the location of all of the strawberries within reach, and then analyse them to determine their ripeness. As such, they need to be located upon the end effector so that they can be within close proximity of each fruit whilst evaluating them.

As the cameras need to achieve both tasks, they need to be at least a minimal distance away from where the strawberries will be picked, so that the focus does not need to be altered. To compensate this limitation, the stereo cameras were placed 150mm away from the nozzle on an actuated pivot (Fig. 72). This lets the cameras rotate down, so that their view can cover their environment or the end effector fruit contact region.

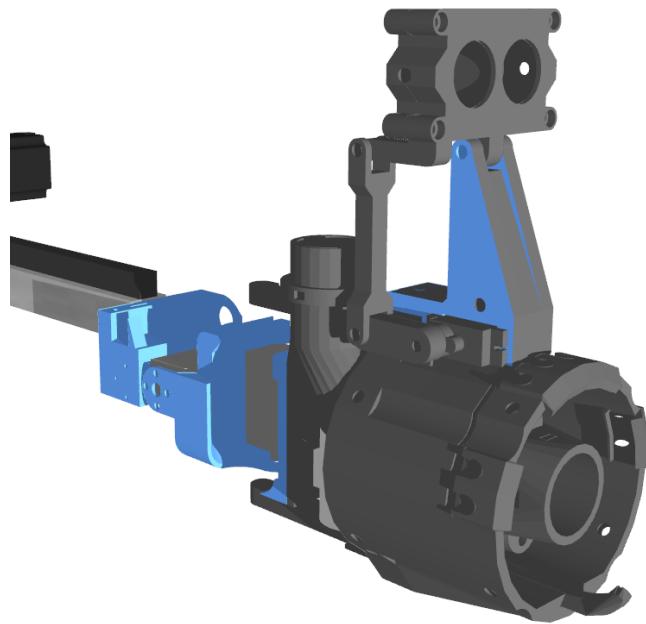


Fig. 72: Camera Positioning

## 11.2.8 End effector Manipulation

To effectively grip any fruit located around the plantation, the gripper mechanism requires as many degrees of freedom as possible. The minimum required on the end of the arm to then allow for a spherical workspace is two; this would enable access from any orientation. However, by introducing more degrees, the gripper becomes more capable. It can plot more complex interception paths that avoid obstacles without the main arm requiring readjustment.

As such, a multi-phase segmented actuator was designed and tested. This mechanism uses a quadagonist/antagonist joint (Fig. 73) set-up to allow for both positional, and displacement control of the head of the actuator.

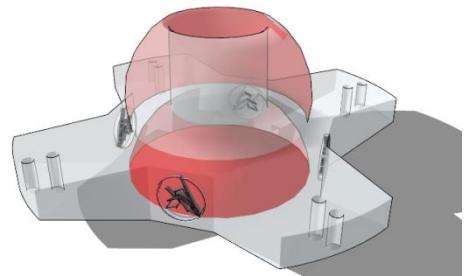


Fig. 73: Quad-Agonist/Antagonist Joint

### 11.2.8.1 Dynamic Displacement Control

Dynamic displacement control is a term that refers to the active manipulation of both the cartesian positioning of an articulated unit, and the spatial size between each of those articulated actuators. By having the ability to actively change the length of the manipulator, this allows for a near infinite number of possible positions and a system that can actively deform to any required shape.

In addition to this dynamic displacement control, by utilising agonist/antagonist joints, they can be co-contracted. Resulting in a system that is able to vary its passive compliance through the use of quadratically expansive tendons. This would mean that the robot would be able to operate safely within the vicinity of human workers, as when the robot approaches said worker, it can reduce the stiffness of its joints. Therefore, if an impact were to occur, there would be a far lower likelihood of injury.

### 11.2.8.2 Design Prototype

The prototype multi stage actuator was printed out of PLA and used braided Nylon acting as the tendons. For testing purposes, it was set up on a small platform and a camera was mounted directly to the top of it. A simple colour tracking algorithm was created to convert the X-Y coordinates of the matched object, to linear movements for the controlling servo motors.



Fig. 74: Multi Stage Actuator

This provided an accurate tracking of the object and was even able to deform due to external forces, whilst maintaining its head's central position. This design also allowed ample room through the very core of the actuator, enabling an in-line vacuum feed for the nozzle. However, such a design is currently only accurately controllable through a direct end feed. Due to the variability in size, shape, twist and orientation, generating an exact kinematic representation of the actuator would require extensive further research and implementation, which on this short-term project, is not available.

As a result, a simpler 2 axis wrist unit was developed (Fig. 75). This wrist unit was designed

to be as small size as possible, with guide channels for all of the wiring that was required. The size of this unit was restricted by the size of the HJ servo motors that were used as active joints. In addition, a dove tail connection was used to mount the wrist unit to the aluminium arm. The reason for this was so that if during testing of the unit, excessive force was caused to the end effector. Rather than breaking the entire unit, the dove tail would instead be the point of failure, reducing the number of parts that would need to be re-manufactured.

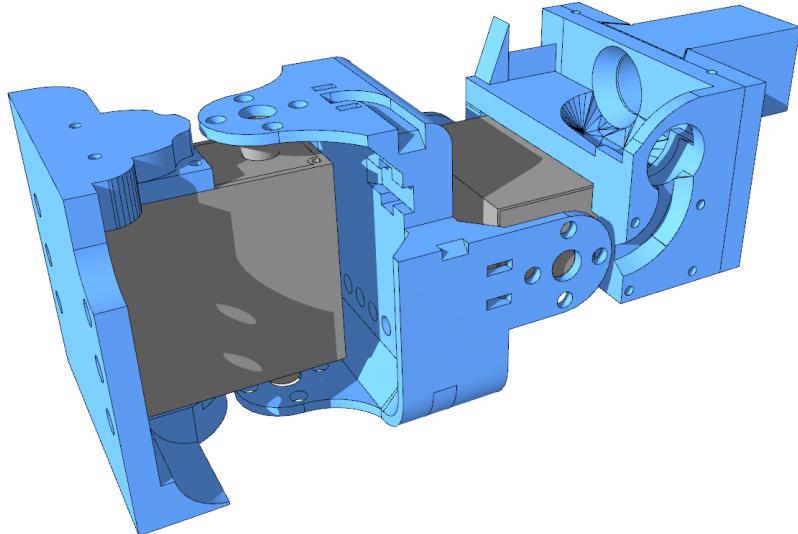


Fig. 75: 2 Axis Wrist

## **12 Final Prototype Design**



Fig. 76: Final Prototype Design

## **13 Programming**

### **13.1 Programming Overview**

The code was split into two main tasks. The one was programming the Arduino Nano to receive commands over a serial connection and to have low level control of the arm. The other was writing a computer program to deal with the required computer vision algorithms and high level control of the arm.

The Arduino Nano is programmed in C++ as well as the computer vision program. The OpenCV 3.2 libraries were used for the computer vision algorithms. Robot operating system (ROS) or Python was not used in an attempt to prevent the code from becoming needlessly abstracted and introducing associated limitations and bugs. Instead C++ (a low-level programming language) was chosen as this allows more control over the program operation and is normally executes quicker than Python.

The Arduino Nano was programmed using the Arduino 1.6.7 IDE and the computer vision software was programmed in Visual Studio 2015.

### **13.2 Code Overview**

This section gives an overview of the code structure and planning to enable a smooth development process.

The program was developed in the Visual Studio 2015 programming environment. Most standard best coding practices were observed, to ensure that the code remained clear and was readable by people other than the developer. This included but was not limited to, well commented code, clear and descriptive variable names, using camel case, proper indentation of the code, using consistent SI units and using a consistent style throughout the project. One team member was responsible for developing the program. This prevented different coding styles throughout the project and complications trying to merge the code developed.

The libraries that would be needed for the project are all included at the top of the project file. Six of the eleven included libraries were for the OpenCV functions and the remaining five were to allow some basic operation with strings, files, input output functions and serial communications.

Following the included libraries were the defined constant values. The OpenCV libraries have the defined constants all in capital letter with underscores separating the words, so to keep a consistent style in the code; all the defined constants were done in capitals with underscores. Using

defined constants makes it much more dynamic to update the code and eliminates the possibility of missing an instance of the values that would be updated. The defined values are mostly serial command numbers, physical dimensions of the system and tolerance thresholds.

The code was declared to be in the “std” and “cv” namespaces. This allowed, functions to be called, and references to code in these libraries made, without the need to explicitly declare the scope each time.

The necessary global variables, Boolean flag and objects were declared. It is often considered bad practice to use global variables however it is justifiable as it makes much of the code simpler.

Functions were used for any pieces of code that would need to be used more than once and for some pieces so that the main code would be more readable. The functions were ordered so that function prototypes would not be required as they would simply use more lines of code.

In the “main” function of the code several initialization processes are performed before the loop begins.

The camera calibration data was read from a file and stored in global variables during the codes initialization processes. This allowed new calibration data to be used via replacing the files, without the need to edit the code. There are two calibration files, the intrinsic camera data and the extrinsic data; they are both in the YAML file format. The calibration maps are generated using the read calibration files. These maps are later used to rectify the camera video streams. While still in the initialization, the video streams are opened, capture resolutions set and image size variable set. The program then moves in to the loop.

The loop begins by capturing a new pair of stereo images, and then proceeds rotate them to the correct orientation and remap them to remove the lens distortion. Algorithms to generate the disparity map and identify the strawberries executed. The identified target are overlaid onto the disparity map; this enables the disparity value to be obtained at the target locations. From the disparity and pixel coordinates the X, Y, and Z coordinates can be calculated relative to the camera. The target coordinate then undergo a transformation to have them represented in the base coordinate system. The target coordinates are then verified to be in range and not already be stored. They are then stored into the global target positions array along with the end effector pose.

The approach vector is calculated to the target in the first row of the target array. The arm is then moved in closer to the target and distance reevaluated and updated in the array. The end effector is then moved to target and grips it, at which point the ripeness estimation occurs. The

targets array is updated with the ripeness coefficient, if the strawberry is not ripe enough to be picked. This prevents the arm revisiting it. A ripe strawberry will have the end effector sheath extend over it and have its stem cut. The picked strawberry will be transported to the storage area and have its data in the targets array removed. This process is repeated for each row of the array.

After all the targets have been picked or determined to be unripe the code will loop back to the beginning and capture a new stereo image pair and search for new targets.

## Code Overview

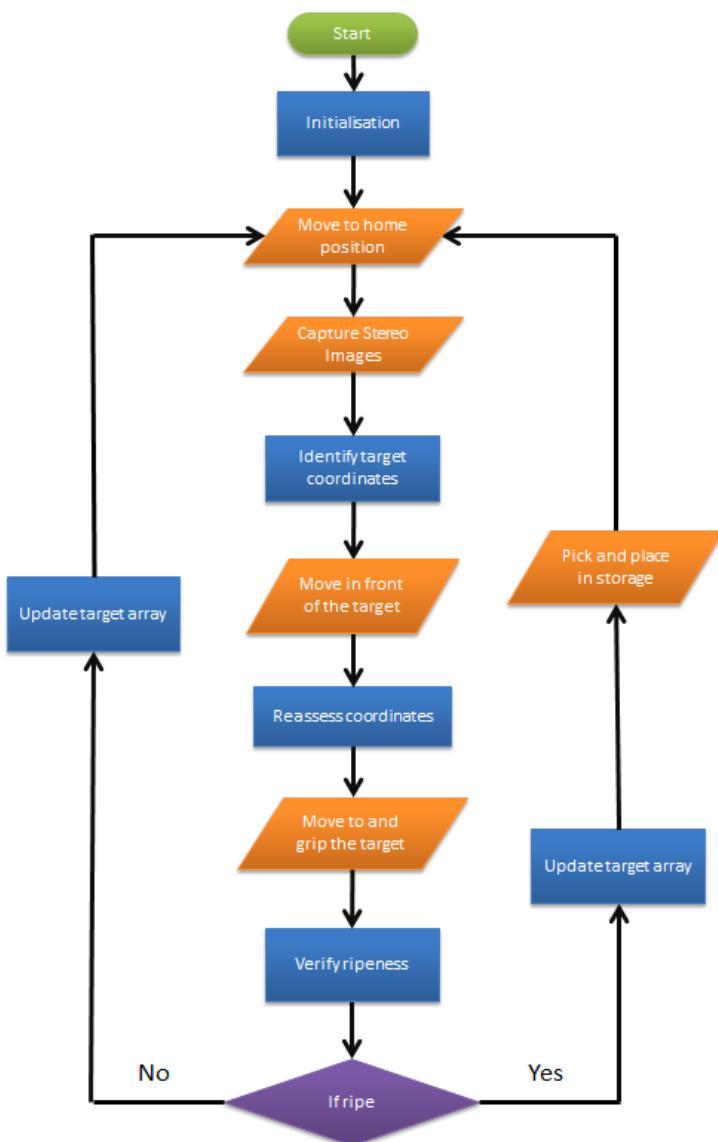


Fig. 77: Code Overview Flowchart

## Ripeness Checking

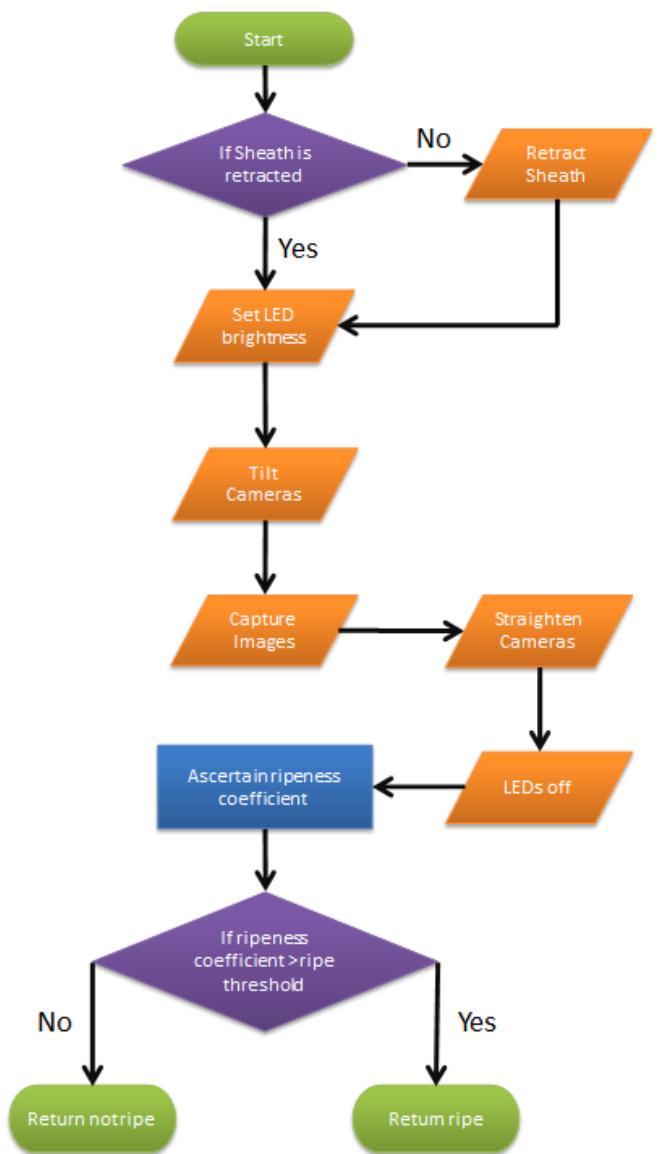


Fig. 78: Ripeness Checking Flowchart

### 13.3 Mode Average Algorithm

The mode average algorithm is a simple but powerful numerical filtering tool, developed to help reduce errors in the disparity measurements. It takes the data set that requires filtering, determines the absolute difference between each of the values, then checks to ensure they are within the maximum allowable variance. The values with the smallest differences are then averaged to give the estimate of the actual value.

### 13.4 Stereo Matching Algorithm

The stereo matching algorithm that was implemented for this project was the semi-global block matching (SGBM) algorithm. This is because it is the best algorithm for stereo correspondence that the OpenCV libraries have. However, it is also the slowest algorithm. Speed is not important for this application so the benefits that a more accurate disparity map will have significantly outweigh the drawback.

The algorithm uses an odd sized pixel block as the block needs to be centered at a pixel value. The pixels are given a cost value that is aggregated by searching the neighboring pixels values, in all eight directions. This will result in the pixels will closely agreeing disparity values being considered the correct match. The algorithm does make the assumption that the surfaces are relatively smooth, but as this is true in most cases it is unlikely to cause errors.

### 13.5 Storing Positions

Once a scene containing strawberries has been captured all of the target positions need to be stored. This prevents a red but unripe strawberry from being continually approached. It also allows all the targets in the entire workspace of the robot arm to be mapped out before any are picked.

A two-dimensional array is used to store the positions of the target, the pose of the end effector when the target was detected and the ripeness if known. Each target has its own row in the array and the columns are in the format of X, Y and Z coordinate of the target, followed by the X, Y, Z coordinate of the end effector then its pitch and yaw. The final column is used to store the ripeness.

As all the coordinates are relative to the base coordinate frame, the orientation of the end effector should not matter.

To add new target coordinates to the array, it must first be checked that they do not already exist in it. As there are slight inaccuracies in the vision system the coordinates are given a tolerance to be counted as the same target. This does give rise to the possibility that two very close targets may be assumed to be the same one. This will be resolved when one of the targets is picked because once picked its coordinates are removed from the array. Therefore, they will not be there to match the second target's coordinates when it is added to the array.

After the end effector has moved to one of the targets and the ripeness has been ascertained, that strawberries ripeness value will be updated in the array. By updating the value, it allows the program to know whether the target has already been checked and not to revisit the same one. If the target is ripe enough then it will be picked and have its row deleted from the array.

The mechanical inaccuracies in the robot arm allow a small but significant movement in the pitch and yaw pivots. For example, 1° of movement in either direction with a target being 1m away could result in a difference of:

$$2 \cdot 1000 \cdot \sin(1) *= 34.9mm$$

Although this is a relatively small discrepancy when compared to the size of a strawberry it is actually quite significant. This requires the tolerances that are used to check if a target is already stored in the array to be larger than desirable.

## 13.6 Serial Communications

To send and receive data between the Arduino Nano and computer/Tinker board USB serial communication was used. Serial communication was setup to use a board rate of 57600 even though the Arduino is capable of higher speeds. This was because it can occasionally miss bits and corrupt the sent byte so it was safer to use a lower speed. Minimal serial data was being sent and it was not time critical so no need to optimize the system for speed.

Data was sent in bytes with one stop bit and no parity bit. DTR control was disabled to prevent the Arduino from resetting every time the port was opened. The Arduino's COM port was defined as com3 in the code however this would depend on which USB port it was plugged into and need to be changed appropriately.

The port was opened whenever data was to be sent or the program was waiting for a response from the Arduino and then closed afterwards. The only data that the Arduino sent to the program was a flag to tell it when a mechanical movement had finished. This allowed the program to send commands at the correct time and capture stereo images when the arm was still.

The first byte sent to the Arduino represents the command number, then the corresponding number of bytes is sent that contain the data that the command need to execute.

The largest numbers that would need to be sent to the Arduino would be the X, Y, Z coordinates of the end effector in mm. Given that the robots workspace is approximately a 1m radius toroidal shape the largest coordinate should not exceed 1000mm. This allows the coordinates to be stored in the ‘short’ data type, which is 16 bits long (2 bytes). Therefore, the maximum value that can be stored in a signed short is given by:

$$2^{15} - 1 = 32,767$$

The short must be split into two bytes to be sent over serial. The eight most significant bits are first taken and stored as a byte then the eight least significant bits are taken and stored. The most significant byte is sent first then the least significant byte is sent. The Arduino receives the most significant byte and shifts it left by eight bits then takes the least significant byte and adds, this then gives the original value of the short.

Angles joint angles are calculated in radians thus require a floating-point data type to store them. This posed the issue of how to send floating points over serial as bytes. The solution that was chosen was to multiply the angle by a large scalar that would result in the enough of the floating points to become decimal then truncate the remaining ones and store this number as a short. The bytes are then split and sent as before. When received by the Arduino the bytes are combined then divided by the scalar to give approximately the original floating point number. This method can introduce a small error but it is negligible compared to the inaccuracies of the joint positions.

The scalar was chosen because of several factors. A complete circle in radians is  $2\pi$  therefore no angle should need to be larger in magnitude than this. The maximum value a signed short can store is 32,767 so the scalar must be small enough not to cause an overflow, thus the maximum value the scalar can be is given by:

$$\frac{32,767}{2 \cdot \pi} = 5,215.03$$

The scalar was set to 5000 because it would be small enough to not cause any overflows and it made debugging simpler.

The commands that were sent to the Arduino are listed in Table 4, although many more commands were implemented on the Arduino, they were not needed in the final version.

<b>Command</b>	<b>Command Value</b>	<b>Required Number of Bytes</b>	<b>Description</b>
Pitch Home	11	0	Homes the pitch axis.
Linear Home	12	0	Homes the linear extension.
Yaw Home	13	0	Homes the yaw axis.
Move Joints	22	10	Moves all the joints to the specified position.
Extend Gripper	23	2	Extends the gripper ring the specified amount.
Led Brightness	24	2	Sets the end effector ring LEDs to the specified duty cycle.
Camera Tilt	25	2	Tilts the camera to the specified angle.
Motor Diver Forwards	26	2	Sets the end effector motor drivers direction and duty cycle.
Motor Driver Backwards	27	2	Sets the end effector motor drivers direction and duty cycle.
Vacuum On	28	0	Switches on the vacuum.
Vacuum Off	29	0	Switches off the vacuum.

Table 4: Arduino Commands

The Arduino waits for a byte to become available in the serial buffer before reading it. The first byte sent and received will be the command number. If the read number matches a command the Arduino will then wait for at least the required number of bytes to become available in the serial buffer before reading them and executing the command. If this command required a movement then after that movement is complete, the Arduino will send the character string “flag” to the back over serial to the program.

## 13.7 Stepper Control

To drive the stepper motors in a smooth and low-wear manner, the delay between pulses is varied based on the arms current orientation, and how far it is in a given movement command.

The speed is adjusted based on the arms current progress though a given move, in order to give a gradual acceleration and deceleration. This stops high torques being generated as the motor tries to overcome the arms inertia. It also helps to stop the arm from oscillating as it stops, allowing more precise movements and helps to maintain the cameras focus.

The first stage of this is to convert its progress in its current movement command to a standardised quadratic.

$$moveSpeed = percentDone > 50 ? \text{round}\left(\frac{(percentDone - 50)^2}{25}\right) : \text{round}\left(\frac{(50 - percentDone)^2}{25}\right)$$

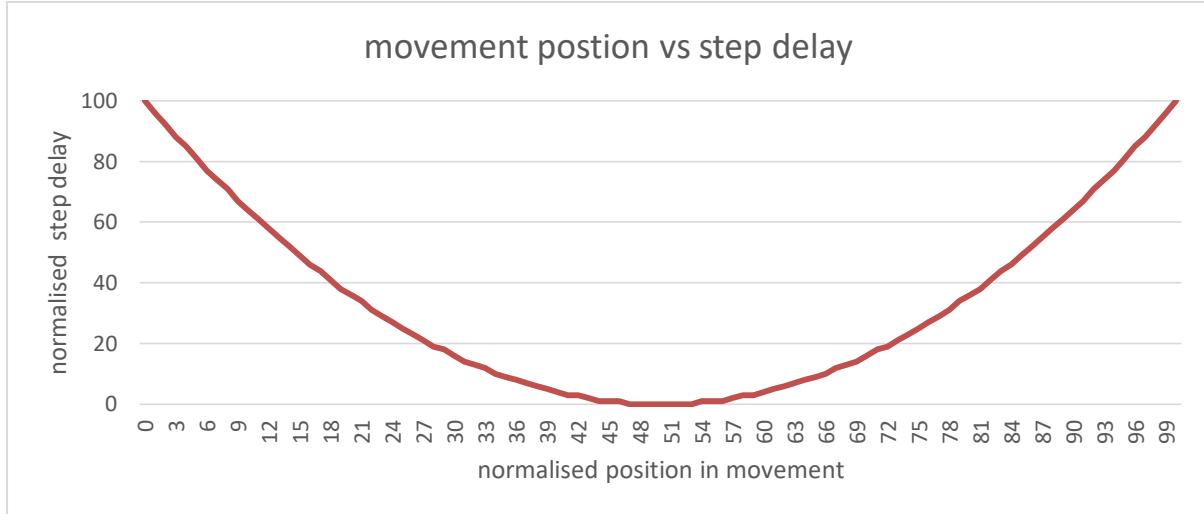


Fig. 79: Movement Position Step Delay

The standardised quadratic is then transformed to lie within a given minimum and maximum pulse time. This allows the motors drive profile to be dynamically modified.

$$targetSpeed = moveSpeed * \left( \frac{\maxPulseTime - \minPulseTime}{100} \right) + \minPulseTime;$$

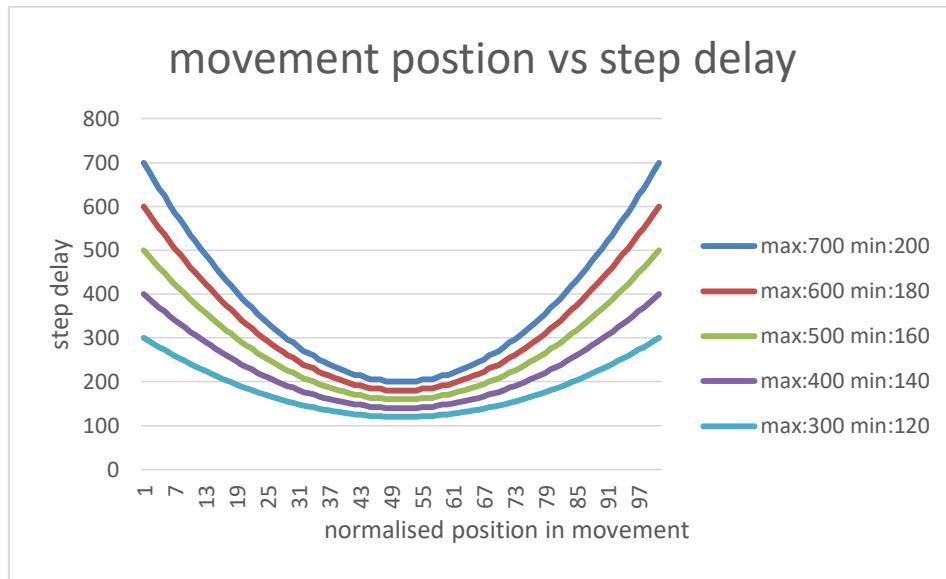


Fig. 80: Movement Position Step Delay 2

The motors drive speed is also modified, as the position of the arm changes, to account for the weight of the arm. This is needed as the torque caused by the weight of the arm changes significantly as the pitch and extension of the arm changes.

$$recommendedSpeed = abs\left(\frac{centerOfMass(pitch, lin)}{7}\right)$$

The centre of mass is based off the two components, on the pitching section, that is not balanced around the pitch axis. These being the linear drive motor and the linear section (units in N/mm).

$$centerOfMass = ((lin - armOffset + pitchSectionLean) * \cos(pitch)) * armWeight + (motorOffset * \cos(pitch)) * motorWeight$$



Fig. 81: Recommended Pulse Times

Minimum and maximum values are then applied to keep the drive rate within a user defined range.

The speed that the arm moves at is then set by comparing the target speed and the recommended speed, and the motors are driven as the slower of the two speeds.



Fig. 82: Movement Position Step Delay 2

## 14 Testing

### 14.1 HJ S3315D Servo Testing

The HJ S3315D Servo's strength was tested by clamping the servo in place just above digital scales. The servo was then driven to its stalling point and the force applied to the scales noted, shown in figure 83. The distance from the servo's pivot to the tip of the bracket is 3.5cm. The rated torque for the servo is 15kg.cm, so the expected force exerted on the scales was calculated to be:

$$\frac{15}{3.5} = 4.286\text{kg}$$



Fig 83: HJ S3315D Test

The measured value was 3.838kg which was slightly less than predicted. This would give the servos a torque of:

$$3.838 \cdot 3.5 = 13.433\text{kg.cm}$$

This is less than the stated torque on the datasheet however still enough to move the end effector with ease.

The HJ S3315D Servo's current draw was tested under three different conditions. The first was under no load, the second was under a 2kg load at 2.5cm from the pivot (figure 84) and the third was the servo being prevented from reaching its target position.

The Servo's current draw under no load is 0.14A. Under the 2kg load at 2.5cm the current draw is 0.46A. While being prevented from moving the current draw is 1.3A.

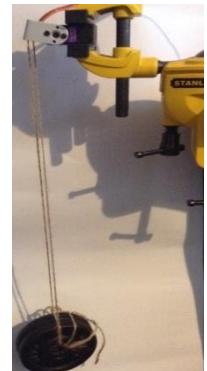


Fig 84: HJ S3315D Current Test

The HJ S3315D Servo's angle accuracy was tested by determining the minimum number of pulse counts that would cause the servo's position to change and using this number to calculate the minimum step angle.

$$\text{Angle Accuracy} = 180^\circ / (\text{450}/\text{Minimum Number Of Pulse Counts})$$

450 is the nominal number of pulse counts for a servo to have in 180° of motion.

The minimum number of pulse counts was determined to be 2 therefore:

$$\frac{180}{(450/2)} = 0.8^\circ$$

The maximum resolution that the HJ S3315D servo can move to is 0.8°

The HJ S3315D Servo's angular velocity was tested by having the servo sweep from its 0° position to its 60° position and timing how long it took to travel that distance. To get accurate timings the movement was filmed in slow motion and played back so the time it started to move and the time it stopped could be noted. This was repeated five times and an average taken (Appendix Table 5).

The servo's maximum angular velocity at 6V is 60° in 0.18 seconds. This is slightly below the quoted 60° in 0.16 seconds.

## 14.2 Micro Servo Testing

The Tower Pro Micro Servo's strength was tested by clamping the servo in place just above digital scales. The servo was then driven to its stalling point and the force applied to the scales noted, shown in figure 85. The distance from the servo's pivot to the tip of the bracket is 1.5cm. The rated torque for the servo is 1.8kg.cm, so the expected force exerted on the scales was calculated to be:



Fig 85: Tower Pro Micro Servo Test

$$\frac{1.8}{1.5} = 1.2 \text{ kg}$$

The measured value was 0.973kg which was slightly less than predicted. This would give the servos a torque of:

$$0.973 \cdot 1.5 = 1.460 \text{ kg.cm}$$

This is less than the stated torque on the datasheet however still enough to actuate the sheath and control the camera angle without any issues.

The Tower Pro Micro Servo's angle accuracy was tested by determining the minimum number of pulse counts that would cause the servo's position to change and using this number to calculate the minimum step angle.

$$\text{Angle Accuracy} = 180^\circ / (450 / \text{Minimum Number Of Pulse Counts})$$

450 is the nominal number of pulse counts for a servo to have in 180° of motion.

The minimum number of pulse counts was determined to be 4 therefore:

$$\frac{180}{(450/4)} = 1.6^\circ$$

The maximum resolution that the Tower Pro Micro servo can move to is 1.6°.

The Tower Pro Micro servo's angular velocity was tested by having the servo sweep from its 0° position to its 60° position and timing how long it took to travel that distance. To get accurate timings the movement was filmed in slow motion and played back so the time it started to move and the time it stopped could be noted. This was repeated five times and an average taken (Appendix Table 6).

The servo's maximum angular velocity at 6V is 60° in 0.136 seconds. This is slightly slower than the quoted 60° in 0.12 seconds.

The Tower Pro Micro servo's current draw was tested under three different conditions. The first was under no load, the second was under a 0.5kg load at 1.5cm from the pivot (figure 86) and the third was the servo being prevented from reaching its target position.

The Tower Pro Micro Servo's current draw under no load is 0.08A. Under the 0.5kg load at 1.5cm the current draw is 0.38A. While being prevented from moving the current draw is 0.8A



Fig 86: Tower Pro Micro Servo Current Test

### 14.3 Linear Movement Speed

The rate of linear extension of the robot arm was tested by having the prismatic joint extend and recording it. It was then analyzed frame by frame to get accurate timings. The joint extends at a rate of 1cm every 5 frames. The camera frame rate was 30fps therefore:

$$1 \cdot \frac{30}{5} = 6\text{cm/s} = 60\text{mm/s}$$

The prismatic joint can extend or retract at a rate of 60mm per second.

### 14.4 Arm Maximum Payload

To test the maximum payload of the arm, a mass was placed on digital scales and then attached to the arm at 300mm from the pitch pivot (figure 87). The pitch motor was driven to attempt to lift the mass. The change in weight that the scales showed is the mass that the arm could lift at this distance.

The pitch motor will lift 6kg at 0.3m from the yaw axis (origin of the base coordinate frame) before stalling.

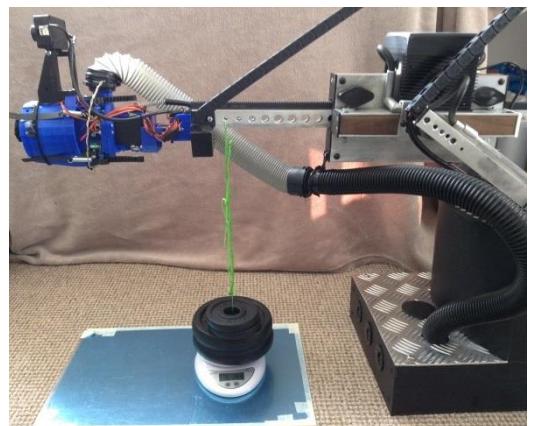


Fig 87: Payload test

The pitch motor therefore can apply a torque of:

$$6 \cdot 0.3 = 1.8 \text{ kg.m}$$

The minimum payloads at displacements from the origin within the robot's workspace are shown in figure 88.

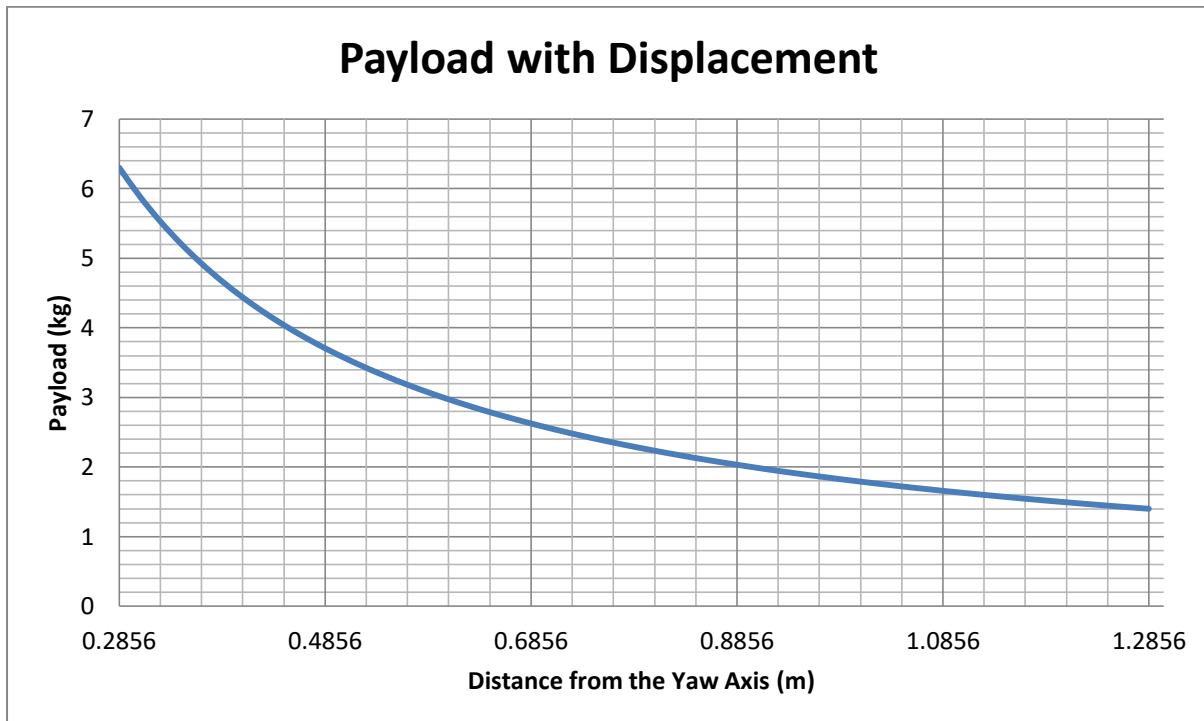


Fig. 88: Payload with Displacement

## **15 Future development**

### **15.1 Drive Improvements**

Following the development of the prototype arm, a direct drive system wasn't the optimal solution; due to the cost/backlash of available motors the arm is capable of large deviations from its intended position.

A large spur or worm gear arrangement (Fig. 89), or a screw drive would allow for a single gearing stage using standard stepper motors, given other concessions that could be made in the design, such an arrangement could be 3d printed or similar low load capacity system. This would also mean that the arm would not be back driveable allowing the motors to assume a lower resting current, improving efficiency and reducing wear on the motors.

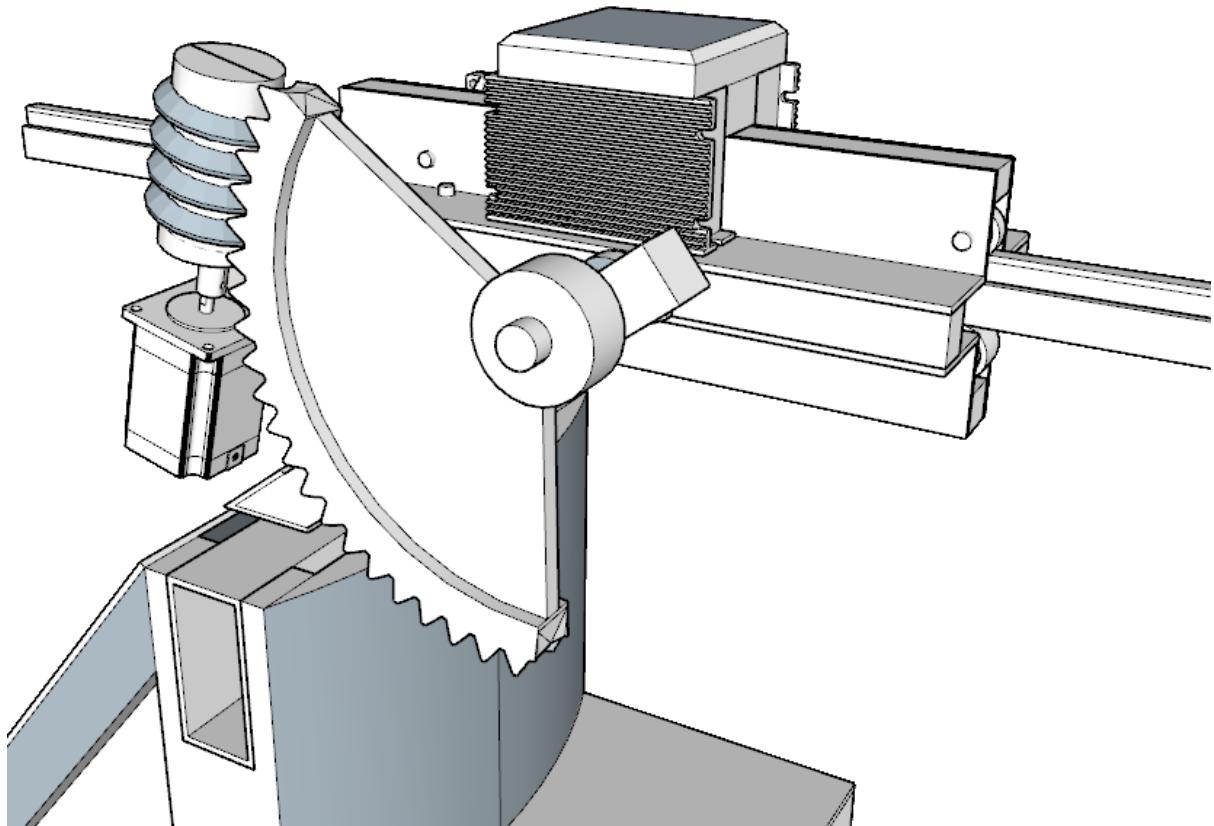


Fig. 89: Worm Gear

Due to time constraints a heavier gauge box section was used for the linear rail; structurally this is more than sufficient for a light duty gripper, and is outright harmful to the arm's performance. Even with attempts made to lighten the structure of the box section it is far heavier than would be required.

Likewise, due to time constraints a much heavier duty rack was used, this also impacted performance. Due to the light duty work a much thinner rack could have been used with negligible impact to the arms structural integrity.

## 15.2 Circuitry Improvements

For the control circuitry, a custom board would be preferred as there are several functions of the tb6000 chip that would be beneficial to have direct access to. Most notably PWM based current control; overheat detection, over current detection and under voltage lock out. A custom board would also be of higher quality than the prebuilt units that were used.

## 15.3 Limit switch Improvements

Given that a production model would be expected to operate in a hostile environment, it would be worth replacing the mechanical limit switches with inductive sensors, to reduce moving parts.

There is also an issue that, depending on what position the arm starts in, the arm may collide with itself or the floor as it attempts to re-establish its home position. This could be solved by adding a low-resolution encoder to the linear section, in order to advise on the order of the homing operation. A one-bit reflection based encoder would likely be sufficient.

## 15.4 Platform Improvements

While the prototype shows promise in its ability to pick strawberries, there is more work needed to have a complete system that can be field tested. The first requirement for making a deployable platform would be building a motorised base, capable of navigating and moving the existing system between plants.

Given the potentially difficult terrain the base would likely consist of a tracked (Fig. 90) or wheeled system that uses GPS, SLAM or radio triangulation to navigate at a field scale, and uses vision processing to navigate between individual plants. Given that most farms do not have a GPRS transmitter nearby, GPS is likely not accurate enough for anything but the most basic of navigation purposes.

The prototype would also need converting to a battery based system. This would not be a significant alteration, as it would only require replacing the vacuum pump with a low voltage variant, as well as adding a battery manager and voltage converters to replace the SMPS.

The biggest issue with running the system on batteries is that a high flow vacuum pump consumes a lot of energy, potentially on the order of a thousand watts. To account for this the base would need to hold a large array of lithium or lead-acid batteries to have a reasonable run time between charges. Alternatively, the system would need a method of switching batteries when they run low, rather than waiting for them to be recharged.

Given that harvesting strawberries is often a rush to get the fruit collected while it is still ripe, a farm typically employs a large number of workers for a brief period of time. While a robot can work longer hours than a human, a single unit is likely not enough to harvest an entire field before a crop goes bad. Because of this multiple units would need to be deployed on a single field to have the same effect. This would mean that the units would need to be able to communicate to keep track of what plants have been harvested, as well as to negotiate moving and charging stations.

## 15.5 Camera Improvements

The cameras used for this project were ultimately the lowest cost cameras available and they were chosen, amongst other reasons, for their availability. This was vital for the research and development of the vision system as it meant a hands-on experience could be quickly learnt. Moving on however, to aid further development to the machine vision, it would be preferable if the cameras were to have an auto white balance disable. This would result in easier frame matching due to a decrease in illumination variances. In addition, a better low light performance would also increase the accuracy of the resulting data as random sensor noise would be far less apparent.

## 15.6 Algorithm Improvements

Over the time span of this project, the developed algorithms have continued to grow in complexity and adaptability. For this project to be carried forward into the future, further development can continue and each aspect of the control can become further refined, allowing for more complex and higher resolution point cloud trajectory planning and subject analysis.

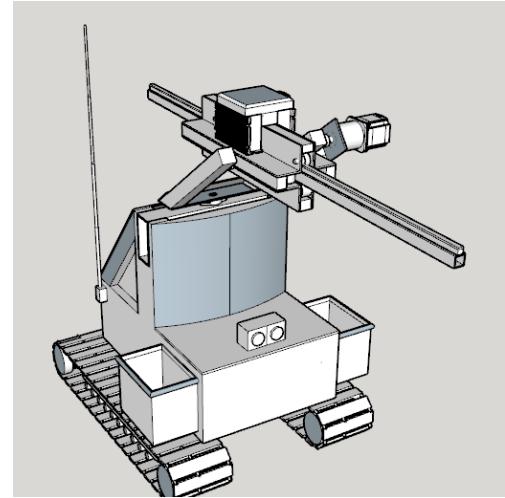


Fig. 90: Tracked Platform

## 15.7 End-Effector Improvements

As this project continues to adapt and become more robust, so must the end-effector. As such, development would revolve around continuing the design and control algorithms for the multi-stage dynamic displacement actuator. Continued efforts would also be made to further reduce the overall size of the head and cutting mechanism, so that even fruits hidden far inside the foliage are harvestable.

## **16 Conclusion**

The final developed prototype functions well and proves that even on a relatively restrictive budget and a tight time frame, when you have a well-coordinated team, even a difficult challenge can be superseded. With the methodology that has been used to approach the issue of automating the harvesting of strawberries, a system has been developed that can identify, analyse and then finally pick the fruit from their unstructured environment.

While there is still more work needed to turn the prototype into a complete field-deployable system, the additions that are needed have already been proven to work, and can be implemented with relative ease. In addition, following on from the patent landscape, the current implementation of the design does not infringe upon any of the existing patents that were analysed, even ones that were flagged for their possible similarities. As a result, there are several unique features that have been developed in this prototype that may be patentable. Most notably, in regard to the functionality of the end-effector, as there currently exist no patent covering the use of a vacuum to manipulate a strawberry from its plantation, or an encompassing drawstring cutting mechanism to sever the strawberry from the stem.

As such, this design could be continued and developed further to ideally be deployed on a larger scale. Helping to not only secure, but potentially revolutionise the agricultural strawberry growing industry.

## 16.1 Expenditure Break-Down

Stage	Item	quantity	Cost
1	Blue PLA Filament	1kg	£28.22
1	RS Pro Flex 45 filament	0.5kg *2	£50.40
1	Chain Tidy	1 meter	£13.60
1	9g Plastic Servo	2	£13.95
1	HJ S3315D 15KG Servo Motor	2	£79.14
1	10A Buck Boost	1	£14.99
2	m1 Gear Rack	1 meter	£33.13
2	m1 Spur Gear (22 teeth)	1 meter	£5.64
2	20mm Aluminium Extruded Strut	1 meter	£5.49
2	6mm Brass Coupler	5	£16.10
2	6mm Flanged Bearings	4	£5.99
2	12mm Flanged Bearings	4	£10.56
2	Nema 23 Stepper Motor	3	£55.00
2	5A Motor Stepper Drivers	4	£51.04
2	Planetary Nema 24 Stepper Motor	2	£123.00
2	8mm Flanged Bearings	4	£11.25
2	1" Aluminium Box Section	2 meters	£29.17
2	8mm Plastic Ball Bearings	16	£10.08
2	13A Tru Connect Plug 1.8m	1	£1.83
2	Inalways 0717-cw Fused Chassis Plug	1	£0.94
2	Mains rated rotation switch head	1	£4.57
2	Mains rated NO switch contacts	1	£3.24
3	Webcams	4	£29.26
3	webcams	2	£16.64
3	Asus Tinker Board	1	£54.99
3	4 channel bi-level shifters	2	£2.99
3	Solid State Relay	1	£5.77

Sum:	
Stage 1	200.3
Stage 2	367.03
Stage 3	109.65

Total:	676.98
--------	--------

## 16.2 Team Dynamic

Working as part of a team has given each member the unique opportunity to co-organise in such a manner to develop in parallel with the other members. With the continued updates delivered between team members regarding one another's current progress, each member has been able to positively support one another, when a difficult challenge has risen. Without this form of co-operation, the successful completion of such a project would not have been feasible. With such a wide range of in-depth knowledge and testing required for every aspect of the build, it was vital that each member was able to specialise in their subject.

## **17 References**

- [1] *Nema 23 Stepper Motor with 15:1 Planetary GearBox Available at: <http://www.omc-stepperonline.com/download/pdf/23HS22-2804S-PG15.pdf>* (Accessed: 28 May 2017).
- [2] *Nema 23 Stepper Motor Available at: [motion.schneider-electric.com/downloads/quickreference/NEMA23.pdf](http://motion.schneider-electric.com/downloads/quickreference/NEMA23.pdf)* (Accessed: 28 May 2017).
- [3] *1" Aluminium box section Available at: [https://www.metals4u.co.uk/aluminium/c1/box-section/c13/25.4mmx25.4mmx1.6mm-\(1x1x16swg\)/p230](https://www.metals4u.co.uk/aluminium/c1/box-section/c13/25.4mmx25.4mmx1.6mm-(1x1x16swg)/p230)* (Accessed: 28 May 2017).
- [4] *m1 Steel Gear Rack Available at: <http://uk.rs-online.com/web/p/gear-racks/8762406/>* (Accessed: 28 May 2017).
- [5] *ST-M5045 2-Phase Stepper Drivers 4.2A Available at: [www.sainsmart.com/vanilla/Uploader/d4/8dffbb1c0f1f4ef5b066d2bee807a9f.pdf](http://www.sainsmart.com/vanilla/Uploader/d4/8dffbb1c0f1f4ef5b066d2bee807a9f.pdf)* (Accessed: 28 May 2017).
- [6] *Arduino Nano Microcontroller Available at: <https://www.arduino.cc/en/Main/arduinoBoardNano>* (Accessed: 28 May 2017).
- [7] *I2C Interface Available at: <http://i2c.info/>* (Accessed: 28 May 2017).
- [8] *I2C 12-bit 16 channel PCA9685 servo driver Available at: <https://cdn-shop.adafruit.com/datasheets/PCA9685.pdf>* (Accessed: 28 May 2017).
- [9] *Pulse Width Modulation Available at: <https://learn.sparkfun.com/tutorials/pulse-width-modulation>* (Accessed: 28 May 2017).
- [10] *SG90 9G Servo Motor Available at: [akizukidensi.com/download/ds/towerpro/SG90.pdf](http://akizukidensi.com/download/ds/towerpro/SG90.pdf)* (Accessed: 28 May 2017).
- [11] *HJ S3315D Servo Motor Available at: <http://www.dx.com/p/hj-s3315d-15kg-torque-metal-gear-digital-servo-for-robot-truck-r-c-helicopter-black-193284#.WSM0bdy1vIU>* (Accessed: 28 May 2017).
- [12] *USB Serial Connection Available at: <https://learn.sparkfun.com/tutorials/serial-communication>* (Accessed: 28 May 2017).
- [13] *ASUS TinkerBoard Available at: <https://www.asus.com/uk/Single-Board-Computer/Tinker-Board/>* (Accessed: 28 May 2017).
- [14] *CANYON CNE-CWC2 Web Camera Available at: <http://canyon.eu/product/cne-cwc2/>* (Accessed: 28 May 2017).

- [15] *Global Population 1927* Available at: <https://www.thoughtco.com/current-world-population-1435270> (Accessed: 28 May 2017).
- [16] *Current Population count* Available at: <http://www.worldometers.info/world-population/> (Accessed: 28 May 2017).
- [17] *United Nations Department of Economic and Social Affairs* Available at: <http://www.un.org/en/development/desa/news/population/2015-report.html> (Accessed: 28 May 2017).
- [18] *What Is Overpopulation* Available at: <http://www.conserve-energy-future.com/causes-effects-solutions-of-overpopulation.php> (Accessed: 28 May 2017).
- [19] *Financial Times: Farms Face Labour Shortage* Available at: <https://www.ft.com/content/a95ecb20-bd66-11e6-8b45-b8b81dd5d080> (Accessed: 28 May 2017).
- [20] *Exploring Consumers' Quality Perceptions of Local NI Food* Available at: [www.niassembly.gov.uk/globalassets/.../rachel-malcolmson-policy-briefing.pdf](http://www.niassembly.gov.uk/globalassets/.../rachel-malcolmson-policy-briefing.pdf) (Accessed: 28 May 2017).
- [21] *Live Science: Crop Failure and Fading Food Supplies* Available at: <http://www.livescience.com/53400-crop-failure-draining-food-supplies-as-planet-warms.html> (Accessed: 28 May 2017).
- [22] *Farmers Test Kinze's Autonomous Harvest System in the Field* Available at: [https://www.agweb.com/article/farmers\\_test\\_kinzes\\_autonomous\\_harvest\\_system\\_in\\_the\\_field/](https://www.agweb.com/article/farmers_test_kinzes_autonomous_harvest_system_in_the_field/) (Accessed: 28 May 2017).
- [23] *Extension: Introduction to Mechanical Harvesting of Wine Grapes* Available at: <http://articles.extension.org/pages/60328/introduction-to-mechanical-harvesting-of-wine-grapes> (Accessed: 28 May 2017).
- [24] *Journal of Agricultural Engineering Research* Available at: <http://www.sciencedirect.com/science/article/pii/S0021863483710206> (Accessed: 28 May 2017).
- [25] *AZCentral: What Is a Strawberry: A Fruit or a Berry?* Available at: <http://healthyliving.azcentral.com/strawberry-fruit-berry-19241.html> (Accessed: 28 May 2017).
- [26] *World Population Growth Over Time* Available at: <http://blog.dssresearch.com/?p=229> (Accessed: 28 May 2017).
- [27] *Skype* Available at: <https://www.skype.com/en/> (Accessed: 28 May 2017).

[28] Autodesk Fusion 360 Available at: <https://www.autodesk.com/products/fusion-360/overview> (Accessed: 28 May 2017).

[29] GitHub Available at: <https://github.com/> (Accessed: 28 May 2017).

[30] Ozaki Laboratory of Utsunomiya University Strawberry Picking Prototype Available at: <http://www.ikegami.co.jp/en/archives/1010> (Accessed: 28 May 2017).

[31] Dogtooth Technologies: Available at: <http://www.dogtoothtech.com> (Accessed: 28 May 2017).

[32] Dogtooth Technologies: Aim Available at: <http://www.dogtoothtech.com/about.html> (Accessed: 28 May 2017).

[33] Strawberry Plants.org: Strawberry Picking Available at: <http://strawberryplants.org/2010/06/strawberry-picking/> (Accessed: 28 May 2017).

[34] Financial Times: Migrants become more picky about British jobs Available at: <https://www.ft.com/content/ffffd7f8c-4456-11e6-b22f-79eb4891c97d> (Accessed: 28 May 2017).

[35] Pips Pick Your Own Farm Available at: <http://www.pipsfruitandveg.co.uk/> (Accessed: 28 May 2017).

[36] Metabolic Profiling of Strawberries Available at <https://academic.oup.com/jxb/article/62/3/1103/479303/Metabolic-profiling-of-strawberry-Fragaria> (Accessed: 28 May 2017).

[37] Atmel ATTiny 85 Available at: [www.microchip.com/wwwproducts/ATTiny85](http://www.microchip.com/wwwproducts/ATTiny85) (Accessed: 28 May 2017).

[38] Raspberry Pi 3 Available at <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/> (Accessed: 28 May 2017).

[39] MG90S Micro Servo Motor Available at: [www.towerpro.com.tw/product/mg90s-3/](http://www.towerpro.com.tw/product/mg90s-3/) (Accessed: 28 May 2017).

[40] Futaba S3010 Standard Servo Available at <http://www.servodatabase.com/servo/futaba/s3010> (Accessed: 28 May 2017).

[41] Dynamixel AX-12A Robot Actuator Available at <http://www.trossenrobotics.com/dynamixel-ax-12-robot-actuator.aspx> (Accessed: 28 May 2017).

[42] Flex 45 Filament Available at: <http://www.dutchfilaments.com/flex-45-filament/> (Accessed: 28 May 2017).

[43] *I2C line state change reference* Available at:

<http://dlnware.com/sites/dlnware.com/files/images/byte-transfer.svg> (Accessed: 28 May 2017).

[44] *ROS: Robot Operating System* Available at: <http://www.ros.org/> (Accessed: 28 May 2017).

[45] *Centrifugal YDK Fan Assembly* Available at:

<http://www.espares.co.uk/product/es1085127/motor-ydk---dc24?pageNumber=1&PartTypeId=1574&ModelId=1084004> (Accessed: 28 May 2017).

[46] *Kudom Opto-Isolated Solid-State Relay KSIM380D10-L* Available at: [http://uk.rs-](http://uk.rs-online.com/web/p/solid-state-relays/1025544/)

[online.com/web/p/solid-state-relays/1025544/](http://uk.rs-online.com/web/p/solid-state-relays/1025544/) (Accessed: 28 May 2017).

[47] *RGB Colour Space* Available at:

[https://people.eecs.berkeley.edu/~sequin/CS184/TOPICS/ColorSpaces/RGB\\_cube.jpg](https://people.eecs.berkeley.edu/~sequin/CS184/TOPICS/ColorSpaces/RGB_cube.jpg) (Accessed 28 May 2017).

[48] *HSV Colour Space Cylindrical* Available at: <http://i.stack.imgur.com/LjbFa.png> (Accessed 28 May 2017).

[49]: *HSV Colour Space Linear* Available at:

<http://docs.opencv.org/2.4/modules/contrib/doc/facerec/colormaps.html> (Accessed 28 May 2017).

[50] *Stereo Coplanar Cameras* Available at: <https://www.intechopen.com/books/current-advancements-in-stereo-vision/3d-probabilistic-occupancy-grid-to-robotic-mapping-with-stereo-vision> (Accessed 28 May 2017).

[51] *Lens Distortion* Available at:

[http://docs.opencv.org/2.4/modules/calib3d/doc/camera\\_calibration\\_and\\_3d\\_reconstruction.html?highlight=sgbm](http://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html?highlight=sgbm) (Accessed 28 May 2017).

[52] *Laser Range Finder* Available at: <https://www.jenoptik.com/-/media/websiteimages/defense-civil-systems/sensors/4x3/dlem/dlem-1k-laser-rangefinder-module.jpg?bc=333333&h=345&la=en-GB&mw=460&w=460&hash=C84B14406DF8A7F61CAF2F66C4979EB28C9D1C3E> (Accessed 28 May 2017).

[53] *Kinect V1* Available at: <https://images-eu.ssl-images-amazon.com/images/G/02/uk-video-games/2010/Xbox/kinectsideways-lg.jpg> (Accessed 28 May 2017).

[54] *Chosen Camera* Available at:

[http://static.nix.ru/autocatalog/webcams\\_canyon/201394\\_2254\\_draft\\_large.jpg](http://static.nix.ru/autocatalog/webcams_canyon/201394_2254_draft_large.jpg) (Accessed 28 May 2017).

[55] *Servo PWM* Available at: <https://sourceforge.net/p/fuyard/wiki/analysis-generic-architecture-hardware/attachment/laboratoire-pic18-servo-pwm.png> (Accessed 23 May 2016).

[56] *Motor Driver L9110* Available at: <https://nvhs.files.wordpress.com/2013/02/datasheet-l9110.pdf> (Accessed 28 May 2017).

[57] NPN Transistor 2N3904 Available at:

<https://www.sparkfun.com/datasheets/Components/2N3904.pdf> [Accessed 28 May 2017].

[58] Ultrabright White LED Available at: <http://www.vishay.com/docs/81159/vlhw5100.pdf> [Accessed 28 May 2017].

## **18 Appendices**

Test Number	Time (s)
1	0.18
2	0.17
3	0.18
4	0.19
5	0.18
Average = 0.18	

Table 5: HJ S3315D

Test Number	Time (s)
1	0.14
2	0.12
3	0.13
4	0.14
5	0.15
Average = 0.136	

Table 6: Tower Pro Micro Servo 9g

PCA9685 Pin	Pin Controls
0	Sheath top micro servo SG90
1	Sheath bottom micro servo SG90
4	Ring LEDs
8	Camera angle micro servo SG90
9	H-bridge pin IA
10	H-bridge pin IB
14	End effector pitch HJ S3315D Robot Servo
15	End effector pitch HJ S3315D Robot Servo

Table 7: PWM Board Pin-out

Arduino Nano Pin	Function	Pin Use
2	Digital Output	Sets the pitch stepper motor direction
4	Digital Output	Enable the pitch stepper motor
6	Digital Input	Detect the limit switches state
7	Digital Output	Enable the linear stepper motor
8	Digital Output	Step the linear stepper motor
9	Digital Output	Sets the linear stepper motor direction
10	Digital Output	Step the pitch stepper motor
11	Digital Output	Step the yaw stepper motor
12	Digital Output	Enable the yaw stepper motor
13	Digital Output	Sets the yaw stepper motor direction
A3	Digital Output	Enables the vacuum relay
A5	I2C SCL	I2C clock line
A4	I2C SDA	I2C data line

Table 8: Arduino Pin-out

Servo	Min Pulse Count	Max Pulse Count
Pitch Servo HJ S3315D	148	611
Yaw Servo HJ S3315D	138	598
Top Sheath Extension Micro Servo	150	550
Bottom Sheath Extension Micro Servo	130	550
Camera Angle Micro Servo	160	280

Table 9: Servo Pulse Range

For further Images, CAD models, code and videos please visit the GitHub repository:

<https://github.com/khaianMarsh/PROJ515-km-jss-is>