Carnegie Mellon University
School of Computer Science
Robotics Department

Detecting and Grasping Sorghum Stalks in Outdoor Occluded Environments

Merritt Jenkins

Submitted in partial fulfillment of the requirements for the degree of
Masters of Science in Robotics, Carnegie Mellon University, August 2017

Technical Report Number: CMU-RI-TR-17-61

Abstract

Conventional methods of identifying and evaluating physical plant traits are labor-intensive and error-prone. For example, plant breeders use a combination of subjective and manual measurements to empirically confirm that new cross-breeds exhibit desired characteristics. Robots offer the opportunity to improve the speed and quality of plant measurements through a combination of computer vision and contact sensing. This thesis describes a custom manipulator and end-effector for field-based contact measurements, as well as online algorithms to visually detect crop stalks in-situ. Field-based crop stalk detection is a challenging computer vision problem due to occlusion by leaves, color similarity between stalks and surrounding foliage, and high stalk density within rows. The hardware and algorithms discussed in this thesis are evaluated in fields of *Sorghum bicolor* in South Carolina, USA.

Acknowledgements

Several individuals have made this research possible:

1. Dr. George Kantor, my thesis advisor, provided invaluable guidance over the last 24 months ranging from manipulator design to point cloud processing. His encouragement and passion for agricultural robotics served as motivation throughout this project.

2. Dr. Srinivasa Narasimhan and Joe Bartels, both members of my thesis committee, provided feedback on my approach to visual detection and systems design.

3. Tim Mueller-Sim developed a highly-reliable robot base, which enabled me to focus my efforts on systems directly related to stalk grasping and detection.

4. Dr. Stephen Nuske and Omeed Mirbod shared images from their custom stereo camera and assisted me in the development of 2D vision algorithms.

5. Dr. Stephen Kresovich, Dr. Barry Flynn, and Zach Brenton provided assistance during data collection in the sorghum fields of South Carolina.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Food and energy security will be global concerns throughout the 21st century. The world population is expected to grow from 7 billion individuals at present to 9.15 individuals by 2050. Much of this growth is expected to take place in regions of already tenuous food security [1]. Climate change will further compound these challenges as adverse weather patterns disproportionately affect subsistence farmers. Crop breeding for traits such as higher yield and drought tolerance is critical to improving the security of food supplies, and will additionally improve energy and climate security. For example, rigorous breeding and management may eventually bring biofuels to price parity with fossil fuels [2], and reduced crop inputs can have a significant effect on greenhouse gas emissions [3]. Improved agricultural efficiency will be critical to global health and economic stability in the 21st century, and plant breeding will play a central role.

The process of developing new crop breeds involves collaboration between both plant breeders and geneticists. Breeders evaluate crops based on physical characteristics, known as phenotypes, and relay that data to geneticists who use it to validate hypotheses and determine future plant crosses. Current methods of evaluating plant phenotypes involve manually measuring crops for specific characteristics such as stalk strength, leaf area, and chemical composition. This laborious process is currently performed by highly skilled plant scientists and breeders who must assess thousands of plants in field conditions. In contrast to the manual processes employed in plant phenotyping, genome research efficiency has improved exponentially over the past decade [4]. This phenomenon is well-recognized by the scientific community, which has deemed it the Phenotyping Bottleneck [5]. Despite the immense agricultural promise of plant genomics [6, 7], optimal breeding decisions are still constrained by manual processes.

Robotics has the potential to transform agricultural phenotyping with consistent, high-throughput data. Indeed, the process of phenotyping indoors (as opposed to outdoors) has been largely automated over the past decade. Companies such as LemnaTec offer end-to-end automation systems including conveyor belts, hyperspectral cameras, and watering systems [8]. Access to vision-based measurements enables breeders and geneticists to make faster and better-informed decisions regarding which plants to cross, in turn bringing improved breeds to market more quickly. However, unlike indoor phenotyping, the pace of automation in outdoor phenotyping has been slow. This is due to the difficulty of state estimation in agricultural environments, the mechanical complexities of all-weather operation, and the computer vision challenges of identifying plant components in unstructured environments. Yet, the confluence of maturing computer vision algorithms, reduced sensor costs, and advancements in prototyping is improving the feasibility of building robots that autonomously navigate, sense, and manipulate their environment.

1.2 Objectives

This thesis explores two challenges in outdoor robotic phenotyping: stalk detection and stalk grasping. Applications for autonomous detection and grasping of plant stalks include yield estimation, harvesting, and strength and composition measurements. Autonomous plant stalk grasping could provide breeders with faster and more reliable access to data at previously unattainable scales. While autonomous plant manipulation has been applied within academia to fruit and vegetable harvesting, these horticultural applications offer differentiable visual cues such as unique color and shape. Detecting a plant organ such as a stalk or leaf and parameterizing its angle and shape is a greater challenge. The methods discussed in this thesis yield the following contributions to the field of agricultural

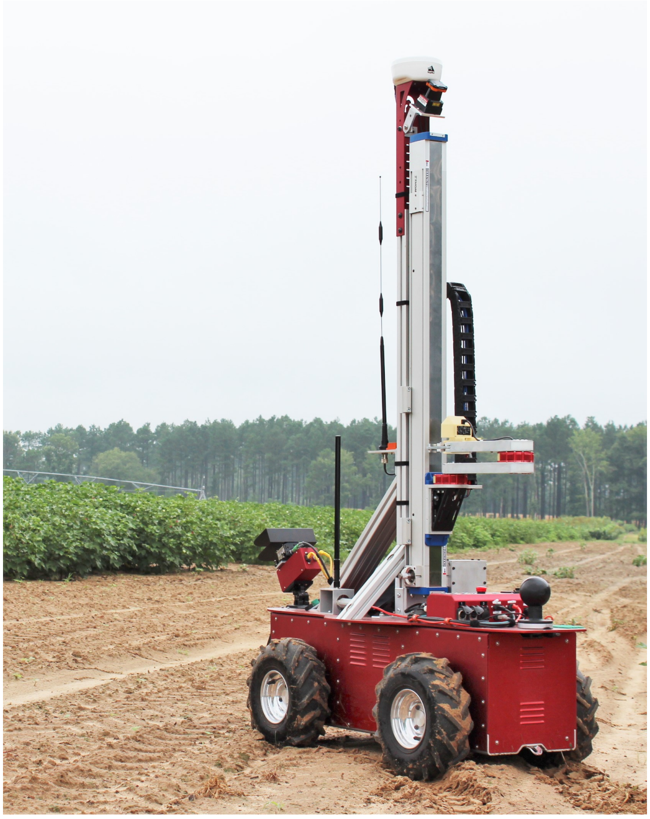


Figure 1.1: The Robotanist phenotyping robot.

robotics:

- Algorithms capable of detecting plant stalks behind occluding features such as leaves
- In-field stalk detection across various field conditions, including lighting and plant maturity
- A manipulator and end-effector capable of deploying a rind penetrometer measurement device in direct contact with plant stalks

The prototype vision and grasping systems were developed specifically for *Sorghum bicolor*, a grain crop similar to corn that thrives on marginal land. A highly diverse plant, there are over 40,000 sorghum accessions housed in the U.S. World Sorghum Collection [9]. This wide inter-accession variation offers tremendous potential for sorghum to be bred for specialized purposes, such as biofuel [2]. Field trials of the prototype system were performed in sorghum fields of Clemson, SC during the summers of 2016 and 2017.

The rest of the paper is organized as follows: Chapter 2 outlines related work in the field, and Chapter 3 provides an overview of the robot base platform. Chapter 4 details the mechanical design of the

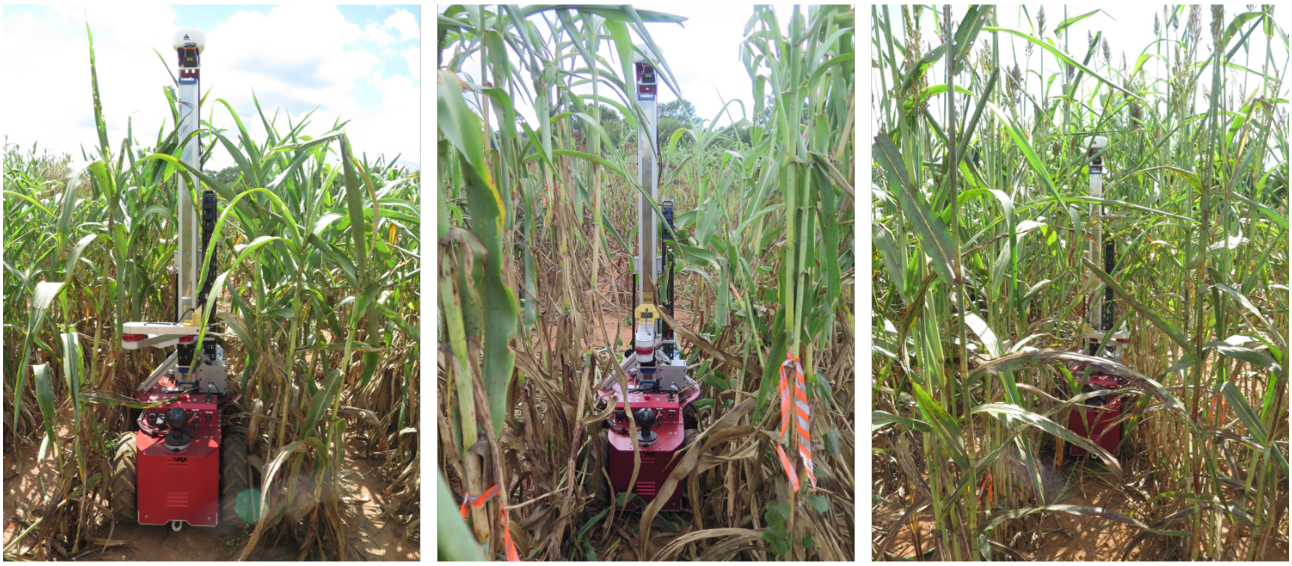


Figure 1.2: The Robotanist navigating sorghum rows.

manipulator and end-effector, and Chapters 5, 6, and 7 describe different stalk detection methods. Lastly, Chapter 8 discusses future work and applications for this novel system.

Chapter 2

Background and Related Work

2.1 Introduction

Research in agricultural robotics began in the early 1980s, but deployment has been largely constrained to academic settings. In a review of the status of agricultural robotics, Muscato et al. note that although the process of harvesting herbaceous crops such as wheat and barley has been highly automated, the process of harvesting horticultural crops is still largely manual [10]. This is due to the complexity of perceiving, manipulating, and navigating variable environments. In an analysis of the challenges to agricultural robotics, Bac et al classify environmental variation into three categories:

- Variation of position, shape, size, and color of objects, as well as occlusion by branches and leaves
- Variation within the operational environment, such as lighting and cultivation systems
- Variation across different types of crops, necessitating custom mechanical designs

The same challenges that constrain robotic crop harvesting apply to outdoor phenotyping, yet both applications may be on the cusp of commercial viability. Blue River Technology and Abundant Robotics, both startups, are proving that computer vision and robotics add value in agricultural settings, and outdoor phenotyping systems may be soon to follow. This review of related work serves to establish a baseline.

A very large proportion of academic literature on agricultural robotics is focused on automated harvest or yield estimation of horticultural crops, specifically fruits, vegetables, and flowers. Detection of plant stalks is often tangent to this main goal [11, 12, 13]. Although some publications focus specifically on plant phenotyping [14, 15, 16, 17], the most widespread research on stalk detection comes from the forestry community. Thus, the perception section of this chapter covers the seemingly disparate fields of horticulture, phenotyping, and forestry.

The manipulation and grasping section of this chapter investigates the challenges associated with designing low-cost mechanical systems that operate reliably in agricultural settings.

Lastly, the control and planning section of this chapter outlines the methods used to navigate manipulators to target locations with vision.

2.2 Perception

The tools and algorithms applied to the research fields of fruit detection, plant phenotyping, and tree stem parameterization differ markedly. Researchers focused on fruit detection almost exclusively utilize 2D images and deploy a toolkit of filters, feature detectors, and Convolutional Neural Networks. Researchers focused on plant phenotyping utilize combinations of 2D and 3D sensors, relying heavily on surface feature-based methods that are robust to noise and occlusion, but computationally intensive. Forestry researchers focus on global 3D methods that can handle large datasets and are perhaps more computationally efficient. The methods applied to each of these applications offer unique advantages that can be leveraged for future improvement.

2.2.1 Fruit Detection in Agriculture

The most common computer vision task in agriculture is fruit detection, which is necessary for yield estimation and autonomous harvesting. Fruit detection algorithms have been refined over the past thirty years on many different horticultural crops, ranging from apples to peppers to grapes, but the field is still an area of active research.

Despite a wide variety of sensors available to roboticists, almost every peer-reviewed detection algorithm exclusively utilizes RGB images. 2D detection and classification is well-studied and computationally efficient, but the computer vision community still struggles with lighting variation in outdoor environments. Sidestepping this challenge, many peer-reviewed publications focus on fruit detection in greenhouses, which offers the advantages of consistent lighting, semi-structured planting, and no wind. The algorithms discussed below focus exclusively on outdoor applications.

The most simple fruit detection algorithms use color thresholding for segmentation. This was shown to be effective by Payne et al., who estimated mango fruit yield by color thresholding the image and performing particle size analysis on the binary mask [18].

Geometric methods add an extra layer of information to image segmentation and may help detect partially occluded target fruits or those of similar color to the surroundings. Silwal et al. used exposure fusion to ensure proper exposure of RGB images, and then detected unoccluded apples using a Circular Hough Transform and partially occluded apples using Blob Analysis [19]. The detection algorithm relied on a black curtain mounted behind the tree, and an RGB camera was calibrated to a time-of-flight camera which provided depth information but was not used for segmentation.

Texture-based methods can be more informative than shape-based methods, as it is difficult to handcraft features that are invariant to all lighting and occlusion scenarios. Pothen et al. describe a method of detecting the reflection of a powerful illumination source on fruit using a custom keypoint detector [20]. The keypoint detector achieved an F1 score of 0.82 on grapes and 0.80 on apples.

Machine learning algorithms can be robust to lighting variation and have only recently been applied to agricultural applications. De-an et al. performed apple detection using a combination of color thresholding, region growing, and shape feature eigenvalues fed into a support vector machine (SVM) with a radial basis function (RBF) kernel. Chen et al. developed a pipeline for orange and apple yield estimation without controlled lighting that consisted of a blob detection fully convolutional network (FCN). Chen and his team achieved an impressive true positive rate of 0.957 and 0.961 for oranges and apples, respectively [21].

FCNs may prove to be a windfall for fruit detection, but 2D methods are still constrained to data with inherently less information than 3D methods.

2.2.2 Plant Stalk Detection in Agriculture

A variety of sensors have been used to detect stalks for agricultural purposes, including laser triangulation scanners [15], uncalibrated digital cameras [11][13], calibrated stereo cameras [12][22], RGB-D cameras [23], and time-of-flight cameras [16]. In contrast to fruit detection, stalk detection algorithms focus on exploiting 3D geometric properties in addition to color information.

Three-dimensional local feature descriptors are a common method of classifying plant organs. For example, Paulus et al. and Sa et al. trained SVMs on surface feature histograms and globally smoothed the output using region growing algorithms [15] [23]. Paulus achieved area-under-curve (AUC) classification accuracy of 0.966, but computation required 5.40 minutes per plant [15]. Sa worked with sparser clouds, achieving a faster computation time (~55 seconds) but a lower AUC of 0.76 [15]. Despite promising results, there is a clear tradeoff between point density, labeling accuracy, and computation time.

Dey et al. also trained an SVM with local feature descriptors, but the descriptor consisted of the relative magnitude of local covariance matrix eigenvalues [11]. The output was then smoothed with a CRF. Although Dey achieved an AUC of 0.92 to 0.96, this was accomplished by generating a single point cloud from 138 8-megapixel images [11].

Chaivivatrakul et al. developed a novel approach to stalk detection in which a 3D plant model is segmented into thin slices, each slice is collapsed to 2D, and least squares ellipse fitting is applied to the 2D slices. The stalk is defined by a sequence of ellipses that overlap within a predetermined threshold [16]. Unfortunately, generation of the 3D model required rotating a potted plant - a method hardly deployable in the field.

Baweja et al. implement a pipeline to detect sorghum stalks that consists of a region-based convolutional neural network (RCNN) and a fully convolutional network (FCN) applied to a single 2D image [24]. The RCNN is trained to detect rectangular regions that contain stalks, and these regions are then fed into the FCN which has been trained to perform pixelwise segmentation. The pipeline takes an average of 0.43s to process each image and yields an r-squared correlation of 0.88 for stalk count.

Of the stalk detection papers discussed, only Dey et al. and Baweja et al. validate a detection algorithm on outdoor field data. Field conditions introduce algorithmic and sensing challenges that include but are not limited to:

- Lighting variation
- Visual occlusion and high plant density
- View-angle limitations due to row-style planting

State-of-the-art crop stalk detection methods are capable of achieving low classification error, but current approaches are not suited to in-field online detection.

2.2.3 Tree Stem Detection in Forestry

The forestry community exclusively uses LiDAR sensors for forest imaging. LiDAR systems such as Terrestrial Laser Scanning (TLS) offer long range and high resolution. For example, Liang et al. use a Leica HDS6100 laser scanner that captures 508,000 points per second. Unfortunately, such scanners are prohibitively costly.

The forestry community is primarily focused on tree volume estimation, and thus a heavy emphasis is placed on stem parameterization. This is achieved exclusively through geometric methods. Kelbe et al. matched cylinders to 2.5D tree reconstructions using 2D line RANSAC on multiple projections, achieving detection rates ranging from 20%-80% [25]. Liang et al. applied a least-squares cylinder fit on 2.5D data, iteratively updating cylinder parameters in 30cm height increments and achieving an overall detection accuracy of 73% [26]. Other methods include using the 7-point RANSAC algorithm to match cylinders to 3D tree trunk point clouds [27], and matching circles to 2D projections of tree trunks using the Hough transform and 2D circle RANSAC [28][29].

Trees are relatively unoccluded close to the ground plane. Thus, many tree stem detection algorithms focus on detection at heights of ~1m [27][25][26][30][28][29]. Stems detected close to the ground

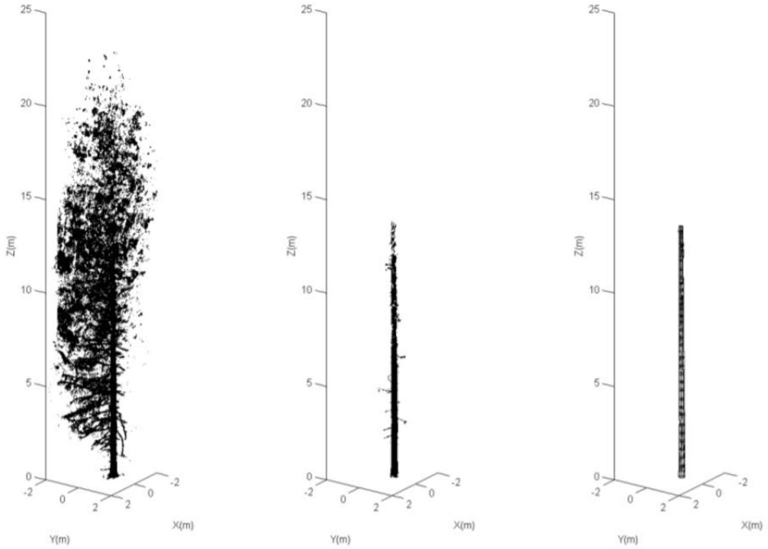


Figure 2.1: Tree stem parameterization from left to right: the original point cloud; the filtered point cloud; tree stem parameterized by cylinders [30].

plane provide initialization points for higher stem segments.

It is clear that the detection methods applied within the forestry community are dramatically different than those developed for fruit detection or crop phenotyping. Although computation speed is not evaluated in the referenced publications, the global methods developed for forestry may offer significant computational advantages that can be leveraged for crop phenotyping.

2.3 Manipulation and Grasping

2.3.1 Manipulators

Robot manipulators are traditionally classified into five categories: Polar, Cylindrical, Cartesian, Jointed-arm/anthropomorphic, and SCARA (Selective Compliant Robot Assembly Arm). Polar, Cylindrical, and Cartesian configurations utilize prismatic joints, which enable highly efficient movements for long reach. On the other hand, anthropomorphic robot arms often have a smaller workspace but often exhibit the ability to move to any position or orientation within that workspace. Anthropomorphic manipulators are useful for servoing to targets parameterized by more than three degrees of freedom, which often require more advanced grasp planning.

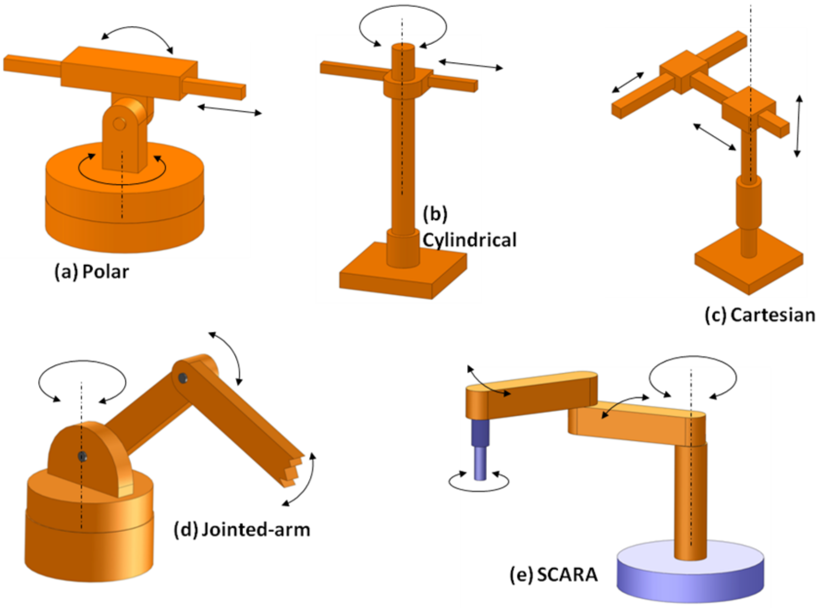


Figure 2.2: Five traditional robot manipulators [31].

Bac et al. surveyed 50 peer-reviewed agricultural robotics papers between the years of 1984 and 2012. They concluded that 78% (35 out of 45) of manipulators were custom-built [32]. While the authors of the survey did not investigate the motivation for custom-built manipulators, it is most likely due to application-specific geometry and weight requirements. Until recently, production manipulators were designed exclusively for heavy-duty industrial tasks, necessitating large motors and heavy structures which are poorly suited for agricultural applications. Baeton et al. tested a VR006L Panasonic industrial manipulator for apple harvesting, but the system required a custom 2D stabilization unit to accommodate the manipulator's high mass.

Scarfe et al. recently developed a custom anthropomorphic manipulator for overhead harvesting of kiwi fruit. The anthropomorphic arm is designed for high speed, extensions greater than 1m, and low cost, so stepper motors located at the base actuate rotary joints through linkages [33]. Other custom long-reach manipulators include those described in [10], [34], and [35]. Irie et al. developed a custom polar manipulator that was capable of reaching deep into asparagus beds for shoot harvesting [36].

While agricultural manipulation tasks requiring a large workspace are still impractical with production manipulators, the advent of small, lightweight manipulators has improved the feasibility of tasks requiring a small workspace, such as greenhouse crop harvesting. Recent papers such as [23] and [37] describe using a Universal Robots UR5 manipulator for sweet pepper harvesting. Van Henten et

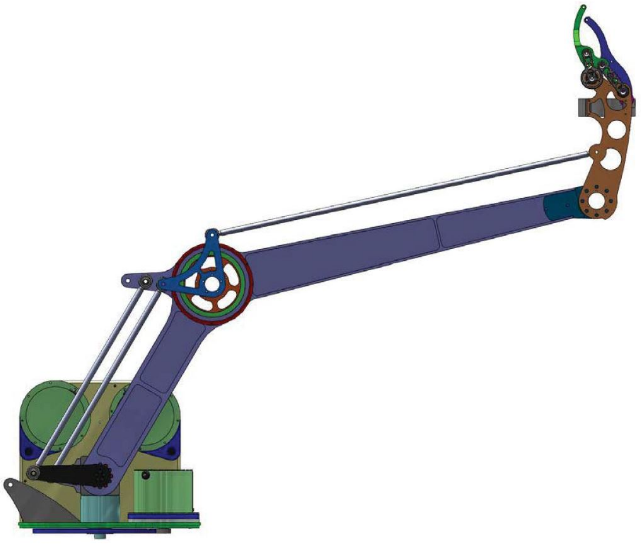


Figure 2.3: A custom arm and two-finger gripper developed for overhead manipulation of kiwifruit [33].

al. describe using a small Mitsubishi RV-E2 for indoor cucumber harvesting [38].

Looking farther into the future, modular systems will enable researchers to easily perform modifications during testing, and perhaps even reconfigure designs for different tasks. Silwal et al. developed a custom 6-DOF manipulator for apple harvesting using off-the-shelf Dynamixel Pro actuators, noting the advantages of a reconfigurable system for research tasks [19].

2.3.2 End-effectors

In addition to surveying agricultural manipulators, Bac et al. evaluated end-effectors described in 47 different papers between 1984 and 2012. They observed that 98% (46/47) of end-effectors were custom-designed for a specific manipulation task. This highlights the lack of off-the-shelf end-effectors that meet agricultural requirements:

- Environmental sealing and concealed mechanisms
- Low weight and low cost
- Capable of grasping highly variable target shapes and sizes

While the survey authors refrain from analyzing the designs of custom end-effectors, it is clear that the vast majority of end-effectors consist of one (or a combination of): fingers, cutting implements,

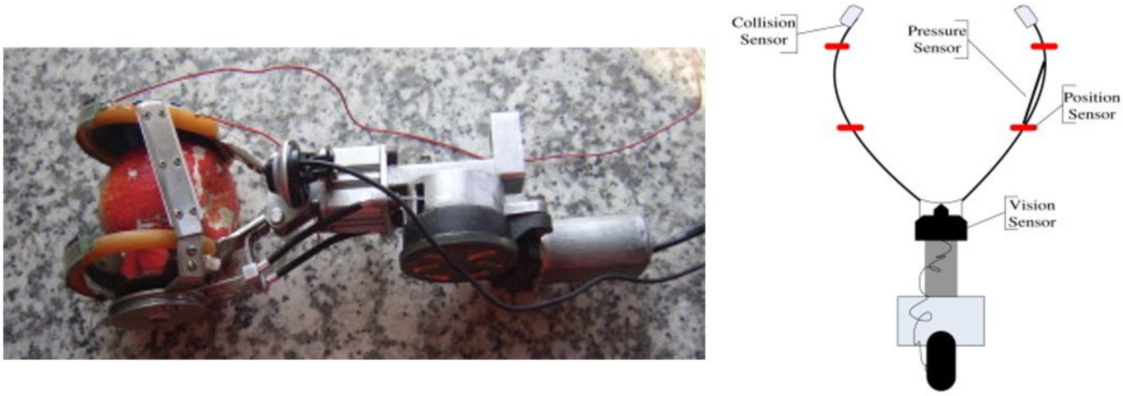


Figure 2.4: A gripper developed by De-An et al. for autonomous apple harvesting [34].

and suction. Although gripper design is dependent upon the application, two-fingered grippers appear to be the most common design.

Muscato et al. developed and tested four different grippers over several years for orange harvesting, ultimately concluding that two independently actuated wire fingers provided the optimal combination of simplicity, light weight, and grasp effectiveness. Muscato's other three prototypes included pneumatic fingers for grasping the fruit, a rotational helix for pulling the fruit, and a multi-fingered tube to fully enclose the fruit [10]. Scarfe implemented a two-finger end-effector design similar to Muscato's on a kiwifruit harvester and proved it to be highly effective [33].

The two-fingered grippers developed by Muscato and Scarfe, while practical, incorporated no grasp sensing. De-An at al. developed a two-fingered gripper for apple harvesting that incorporates microswitches for collision avoidance and pressure sensors for grasp detection [34]. Hayashi et al. similarly incorporated a photoelectric sensor on a 2-DOF gripper to estimate the diameter of eggplants during grasping [39].

Suction gripping offers advantages for harvesting fragile fruit with thin skins. Baeten et al. developed a custom silicone suction gripper to gently grasp apples [40], and Hayashi et al. developed a suction mechanism for strawberry harvesting [41]. However, pneumatic suction gripping does not appear to offer advantages over finger-based mechanisms when the target object is durable and of a known shape, such as a stalk. Silwal et al. note that, unlike vacuum and funnel designs, fingered grippers incorporate the ability to add grasp planning in cluttered environments [19].

A wide variety of end-effector designs have been tested over the past 30 years, and the agricultural robotics community appears to be converging on a consensus that mechanically simple, lightweight grippers offer superior performance.

2.4 Control Architecture and Motion Planning

Sensing is critical to manipulation within unstructured environments. Sensor configurations drive high-level control architectures, and most agricultural manipulation systems exclusively rely on vision-based control. Vision-based control can be segmented into two arenas: visual servo control and look-and-move control. Both types of servo control have been tested in outdoor applications.

Visual servo control entails detecting an object of interest with a camera and using feedback information from that camera as the end-effector is traveling to the object. Visual servo control can be broken into two main types: image-based servo control and position-based servo control. In image-based servo control, the error signal is defined solely by image features rather than an estimation of the target object's pose. Position-based servo control, on the other hand, requires estimation of the target object's 3D pose with respect to the camera [42].

Look-and-move control involves detecting objects within the world or end-effector coordinate frame and moving the end-effector to a desired location without any update from the vision sensors. Look-and-move control offers advantages when target detection and position estimation are computationally intensive.

Most agricultural manipulation systems utilize image-based visual servoing because of the speed and simplicity of the control architecture. 2D image segmentation is well-researched, and the image Jacobian is fast to calculate. For example, Barth et al. used a single RGB camera and a blob detector to segment red peppers, which fed into an image-based visual servo architecture that positioned the end of a Baxter robot arm relative to the center of mass of red peppers [43]. Motion control of the arm was implemented through Visual Servoing Platform (ViSP), a ROS plugin, and pose estimation was accomplished at 5-10Hz.

De-an at al. note that image-based visual servo control inherently avoids obstacles within the camera's field of view, which greatly simplifies control [34]. However, image-based visual servoing alone may be insufficient to properly position an end-effector relative to an object, as agricultural targets are often highly occluded. Two Japanese teams combined image-base visual servo control with fuzzy logic to better approximate the optimal target region of eggplants relative to an end-effector [44, 39]. Another Japanese team focused on detecting a strawberry's penduncle [41], which is a smaller target for image-based visual servoing and thus potentially less prone to error.

Hybrid visual servoing methods offer more robust pose estimation and eliminate some of the challenges associated with image-based servoing, such as camera retreat. Mehta et al. developed a visual servo algorithm for robotic orange harvesting that utilizes one fixed camera with a global view of the tree and a second camera positioned in the hand of the gripper [35]. The system combines Proportional-Derivative (PD) image-based servoing with distance information estimated from the two cameras.

Unlike the described papers which all use an eye-in-hand 2D camera as the primary sensor, Silwal et al. used a time-of-flight sensor and RGB camera mounted rigidly to the base of a robotic arm. Rather than visually servoing to the target, Silwal used open-loop look-and-move control and a linear trajectory to position the end-effector, achieving an impressive 85% picking efficiency.

As is the case with many aspects of agricultural robotics, there are trends within the field but the community is far from consensus. Perception, grasping, and control in agricultural robotics are all still areas of active research, and best practices are sure to evolve as new sensors and hardware become available.

Chapter 3

Robot Platform Overview

The Robotanist is designed as an experimental, modular platform for sensor and algorithm development. Computer and sensor mounts are designed to be reconfigurable, and the vehicle can transport a variety of payloads. Although designed for sorghum phenotyping, the Robotanist platform is capable of navigating a number of different crops. This overview describes the capabilities of the platform and the design decisions involved in its development.

3.1 System Requirements

Robot system requirements are primarily commanded by crop spacing and geography. Sorghum is traditionally grown in rows spaced at 30", so it was critical that the robot width meet this constraint. In order to fulfill the requirements of plant breeders and geneticists, the robot must be able to traverse 1-2 hectare breeding plots within several days. This temporal requirement was necessary to ensure that plants measured at the beginning of a robot's traversal had not grown significantly by the end of the traversal. In order to meet this temporal requirement, the robot had to be capable of a 1.5 m/s velocity and operate continuously for eight hours on a single battery charge.

Sorghum thrives in hot, humid environments, and South Carolina summers can reach above 40° Celsius. Summer temperatures compelled the maximum operational temperature requirement of 45°

Celsius. Fine soils and frequent thunder storms necessitate a robot that is resilient to dust and water.

Other performance requirements were more general to robotics rather than specific to Sorghum breeding plots. These include the ability to transport a 50kg sensor payload, wirelessly communicate with a base station up to 500m away, be teleoperable, and be capable of turning within a radius of <2m.

3.2 System Architecture

3.2.1 Mechanical Systems

The robot base is 1.34×0.56×1.83m (L×W×H), which leaves approximately 0.10m of clearance on either side of the base when traveling through crop rows. The total robot mass is approximately 140kg, of which a significant portion is accounted for by a 24V, 100 Ah LiFePO4 battery.

The robot base is actuated by four independently-driven 200 W brushless DC (BLDC) motors in a skid-steer configuration. The BLDC motors are connected to 50:1 gearheads and are capable of propelling the robot at 2.0 m/s. The motor drivers communicate over Ethernet and control the motors with a manually tuned proportional-integral-derivative (PID) controller.

Communication between the motor controllers and other sensors is controlled by three Intel NUCs. The majority of sensors are controlled over Ethernet, and all three NUCs utilize Robot Operating System (ROS) for communication. Each NUC contains a 3.4 GHz Intel i7 processor, 16 GB RAM, and a 1 TB SSD storage drive.

Robot sensors are powered by four DC-DC converters that convert 24V unregulated battery power to 24V regulated, 12V regulated, and 5V regulated power.

3.2.2 Pose Estimation

Three sensors are used to estimate the robot's pose in the field: a global positioning system (GPS), an inertial measurement unit (IMU), and wheel encoders. The GPS system is Real Time Kine-

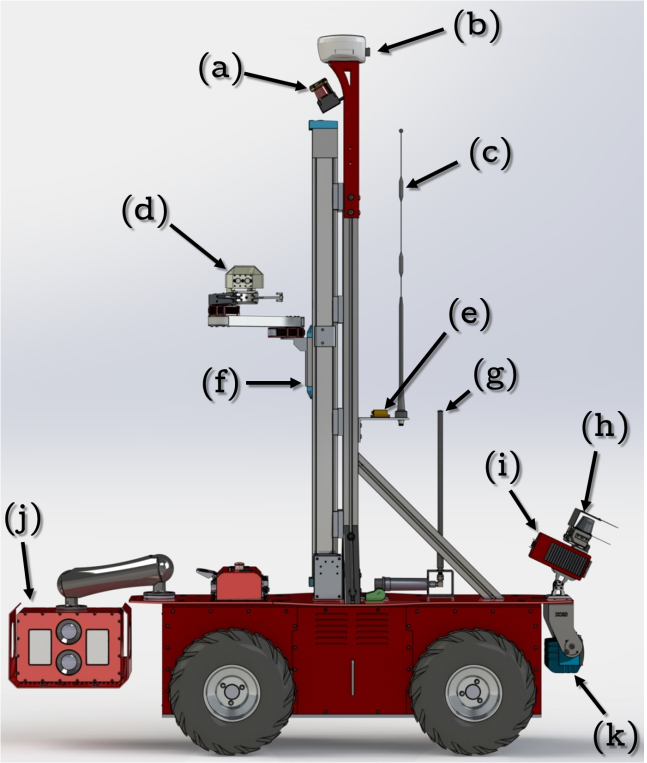


Figure 3.1: A diagram of sensors mounted on the Robotanist: a) Planar LiDAR; (b) GPS receiver; (c) 900 MHz antenna; (d) Custom manipulator; (e) IMU; (f) Linear stage for mast sensor payload; (g) 2.4 GHz antenna; (h) Planar LiDAR; (i) RGB camera; (j) Custom stereo camera with active lighting; (k) Time-of-flight camera [45].

matic (RTK), which uses a base station to measure the phase of the signal's carrier wave to provide centimeter-level accuracy to the receiver. The receiver is a Novatel SMART6-L, which is mounted to the robot at approximately 2m above ground level. The height of the GPS receiver is intended to minimize signal attenuation as the Sorghum plants grow and provide a clearer line of sight to the GPS base station. The GPS base station is installed approximately 2km away from the breeding plot and signals are broadcast over 900MHz radio.

The height of the GPS receiver complicates state estimation because a small angular change of the robot base can cause a large error in GPS positioning accuracy. This issue is magnified as the robot travels down a sorghum row because the mast oscillates at approximately 1 hz. To overcome this issue, an XSENS MTi-30 Attitude and Heading Reference System (AHRS) is installed on the robot mast. This sensor includes a factory-tuned Extended Kalman Filter (EKF) to fuse the accelerometer, gyroscope, and magnetometer data.

Vehicle state estimation is performed using an Unscented Kalman Filter (UKF). The UKF fuses data from the wheel encoders, the AHRS, and the GPS to provide an estimate of the robot state in a fixed coordinate frame.

3.2.3 Navigation

Autonomous navigation is primarily achieved through an implementation of the Pure Pursuit algorithm with a fixed look-ahead distance [46]. In order to log GPS waypoints that are used as inputs for path following, the robot must first be teleoperated through sorghum rows.

Several other sensors are used for autonomous navigation research: a forward-facing time-of-flight (TOF) camera, forward-facing planar LiDAR, and two RGB cameras. The TOF camera (SICK Visionary-T) is used for obstacle detection, while the RGB cameras (IDS UI-5240CP-C-HQ) and planar LiDAR (Hokuyo UTM-30LX and SICK TiM561) are used for improved in-row state estimation.

3.2.4 Visual Phenotyping

A side-facing custom flash camera as described in [20] and shown in Figure 3.2 is mounted to the rear of the robot and operates at 5Hz. The active illumination serves to highlight the foreground, which aides in stereo reconstruction and feature detection. The baseline of the camera is 120mm, and the sensor resolution is 9MP. The camera is connected to a computer inside the robot base over USB 3.0, which is critical for transmitting large images at high frame rates.

3.3 Mission Profile

Field testing in South Carolina was performed daily over the course of four one-week periods. Outdoor temperatures ranged from 31-38° Celsius, and the robot was intermittently caught in rain showers. Navigation consisted of robot teleoperation when gathering images with the custom stereo camera, and GPS waypoint following when testing autonomous navigation. The robot was operated for

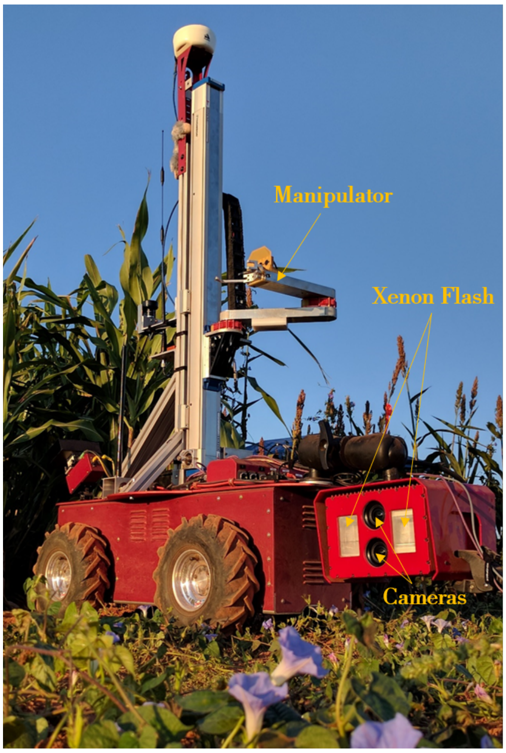


Figure 3.2: A custom side-facing stereo camera for phenotyping.

approximately 8 hours each day in the sorghum breeding plot. Robot travel totaled 44.7 km by the end of the first summer. Data collection included stereo images, LiDAR point clouds, 2D images, wheel odometry, and GPS position, and was recorded using the rosbag package. This ROS package records and time stamps all messages being published by sensors on the robot.

Chapter 4

Manipulator and End Effector Design

The goal of the robotic manipulator is to detect sorghum stalks, servo to their location, grasp them with a gripper, and deploy a rind penetrometer measurement from the end-effector. A rind penetrometer traditionally consists of a needle 15mm in length and 3mm in diameter attached to a force gauge, and is described in section 4.5. The penetrometer measurement is useful to plant breeders and geneticists because it provides an estimate of stalk strength - an important breeding trait. This chapter details the components required to autonomously perform the penetrometer test, including sensors, mechanical design and analysis, and controls, and concludes with an analysis of field test results in South Carolina.

4.1 Sensor Selection

A variety of sensors were evaluated for stalk perception, ranging from consumer-level RGBD cameras to high-quality time-of-flight cameras. The main sensors evaluated are presented in Table 4.1. The primary limitations encountered during sensor selection were minimum range, depth accuracy, and outdoor operation. Sorghum crop row spacing is approximately 760mm, so the center of the robot is positioned approximately 380mm from a row's center. This tight row spacing, combined with the fact that sorghum stalks tend to lean out from the row's center, informs the requirement of a 250mm minimum sensing range. This constraint eliminates several stereo cameras from the list of options, such as the FLIR BumbleBee series and the Stereo Labs ZED camera, because they exhibit a wide baseline and a minimum range of approximately 500mm. Some LiDAR sensors are capable of detecting objects at very short ranges but exhibit high statistical error in depth estimation. For example, the SICK TiM5 series has a statistical error of 20mm, which is greater than the diameter of most sorghum stalks. On the other hand, laser triangulation sensors such as the SICK OD2 offer sub-mm depth estimation at short ranges but emit only a single beam. Sensors that rely on projected infrared light, such as the Microsoft Kinect, do not perform well outdoors in full sunlight. Given the constraints outlined above, off-the-shelf sensor options are limited to the following: MultiSense S7 Stereo Camera, DUO3D MLX, and PMD time-of-flight sensors.

Table 4.1: Imaging Sensors and Specifications

Imaging Sensors	Min. Range (mm)	Approx. Cost	Resolution (MP)	Outdoor Capability	Max. Frame Rate (fps)
PMD Camcube 3.0	300	$25,000	0.04	Yes	40
Microsoft Kinect	800	$200	0.3	No	30
PMD CamBoard Nano	100	$900	0.04	Unclear	45
Point Grey Grasshopper3	100	$975	3.2	Yes	121
DUO3D Stereo Camera	100	$700	0.3	Yes	15
ZED Stereo Camera	500	$450	5.5	Yes	15
Bumblebee2 Stereo Camera	500	$2,400	0.8	Yes	20
Bumblebee XB3 Stereo Camera	500	$3,500	0.8	Yes	20
Multisense S7	200	$7,000	4	Yes	10
Velodyne Puck	1000	$8,000	0.3*	Yes	1*
SICK TiM Series Planar LiDAR	50	$2,000	0.0003	Yes	15
SICK OD2 triangulation sensor	50	$1,000	0.000001	Yes	2000
Custom Stereo Camera	250	$5,000	9	Yes	5

*300,000 pps

The tradeoffs between a time-of-flight (ToF) camera and a stereo camera are less straightforward. Stereo cameras can produce much higher resolution images, but the actual resolution in 3D is a function of correspondence matching. Light information can be useful when differentiating young stalks from leaves, which is not available from ToF cameras, but ToF cameras are insensitive to changes in lighting conditions. Although time-of-flight sensors offer the advantage of full-resolution 3D images at every frame, pixel intensity and high resolution were deemed to be more important. Thus, the sensors investigated in this project are the DUO3D MLX, a custom stereo camera, and the Carnegie Robotics MultiSense S7. These sensors are discussed in Chapters 5, 6, and 7 respectively.

4.2 Manipulator Design

A limited number of manipulator requirements compelled the majority of manipulator design decisions. These requirements are outlined in Table 4.2. The manipulator height requirement of 1.5m is motivated by a future desire to take data from sorghum leaves and perform visual scans of the entire sorghum plant. The manipulator must also be capable of reaching plants on both sides of the row, draw minimal power, not cause significant base platform instability, and be relatively low cost.

Table 4.2: High-Level Manipulator Requirements

Height	Reach 1.5m high into the canopy
Reach	Grasp plants >0.5m away
Power	Less than 100W to operate
Mass	Less than 25 kg
Speed	Move a 2kg end-effector mass at 1 m/s
Sealing	Full-time outdoor operation
Cost	Less than $20,000

Several off-the-shelf anthropomorphic manipulators were evaluated for feasibility and outlined in Table 4.3. Unfortunately, none of these manipulators met the specifications of height, cost, mass, and environmental sealing. The cost and performance specifications of the Kinova JACO2 and MICO2 are compelling, but the low IP rating limits these models to indoor operation. The GearWurx ARM 3.0 meets the overall specifications, but the system uses hobby-grade RC motors which may be unreliable. The AUBO i5 robot meets every specification except for height and power, and would have been a viable option but it was not commercially available at the time.

Table 4.3: Manipulator Performance Specifications

Companies	Manipulator	DOF	Reach (mm)	Mass (kg)	Payload (kg)	Cost	Power (W)	IP Class
ABB	IRB120	6	580	25	3	$30,000	240	30
Barrett	WAM	4	1000	25	4	$160,000	30	Unspecified
Kinova	JACO2	4	750	3.6	4.4	$22,950	25	X2
Kinova	JACO2	6	900	4.4	2.6	$29,950	25	X2
Kinova	MICO2	4	550	3.8	5.2	$16,900	25	X2
Kinova	MICO2	6	700	4.6	2.1	$20,900	25	X2
Robai	Cyton Gamma 300	7	480	1.2	0.3	$5,000	24	X2
Universal Robots	UR3	6	500	11	6.6	$23,000	100	54
Universal Robots	UR5	6	850	18.4	5	$35,000	150	54
Universal Robots	UR10	6	1300	28.9	10	$45,000	250	54
GearWurx	ARM 3.0	6	2200	9.25	4.5	$5,995	90	Unspecified
Robotis	Manipulator-H	6	633	5.5	3	$18,900	640	54
Schunk	LWA-4D	7	1113	18	10	$75,000	120	54
AUBO	I-5	6	924	24	5	$15,000	200	54

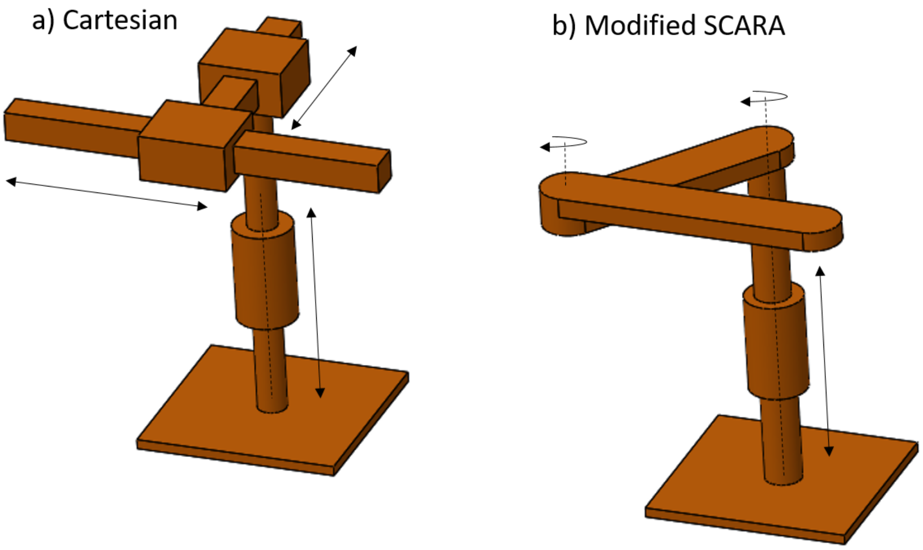


Figure 4.1: Custom manipulator architectures.

Given the fact that no off-the-shelf manipulators met our specifications, custom manipulator designs were investigated. Sorghum stalks are roughly cylindrical and grow vertically, and the desired manipulation task is agnostic to location on the stalks' circumference. Thus, only three degrees of freedom are required to position an end effector relative to a sorghum stalk. The two manipulator configurations that were considered are shown in Figure 4.1. Both the Cartesian and Modified SCARA manipulators exhibit extremely simple inverse kinematics. The Modified SCARA manipulator design offers the additional advantages of lightweight, low-volume rotary joints, and it is these advantages that led to its eventual selection as the ideal manipulator design for stalk grasping.

The first axis of the Modified SCARA manipulator is a prismatic joint, so the inverse kinematics (IK) simplify to that of a planar RR manipulator. The IK can be solved with elementary trigonometry using the conventions outlined in Figure 4.2.

The joint lengths L_1 and L_2 are known, and the (x, y) location of the target point is also known. The distance from the origin to the target point is defined as r. The variables θ_1 and θ_2 represent the desired joint angles, and α is the supplementary angle of θ_2. The law of cosines can be used to solve for α:

$$r^2 = L_1^2 + L_2^2 - 2L_1L_2cos(\alpha)$$

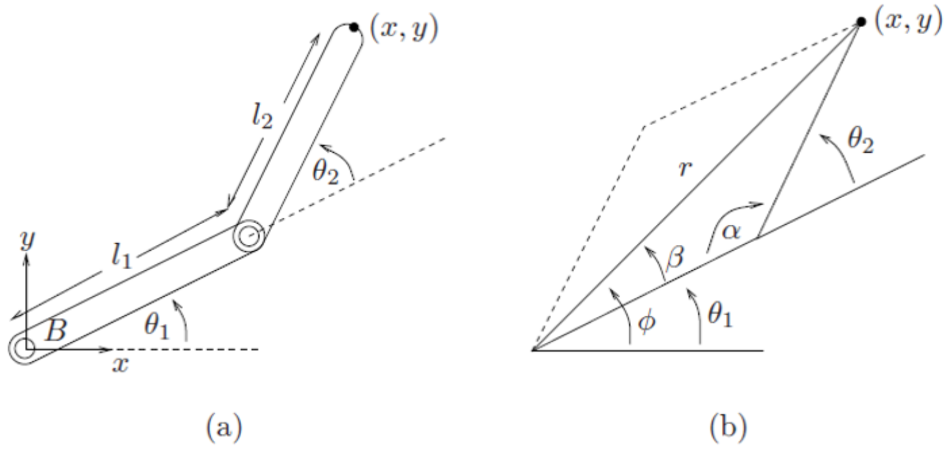


Figure 4.2: Two-link planar RR manipulator: a) Links and desired angles; b) Two elbow configurations and other variables critical to solving the inverse kinematics [47].

$$\alpha = cos^{-1}\left(\frac{L_1^2 + L_2^2 - r^2}{2L_1L_2}\right)$$

The angle of the second joint, θ_2, is defined as $\theta_2 = \pi \pm \alpha$. The two values of θ_2 represent the two configurations of the elbow. In application, the elbow is always in the left configuration when grasping stalks to the left of the robot, and vice versa.

The angle ϕ is defined by the target points (x, y) where $\phi = atan2(y, x)$. The target angle θ_1 is defined as the difference between ϕ and β, where β is the angle between the triangle hypotenuse r and the first joint L_1. The angle β can be solved using the law of cosines:

$$L_2^2 = L_1^2 + r^2 - 2L_1rcos(\beta)$$

$$\beta = cos^{-1}\left(\frac{L_1^2 + r^2 - L_2^2}{2L_1r}\right)$$

The full solution to solving for θ_1 and θ_2 is shown below:

$$\theta_1 = atan2(x, y) \pm cos^{-1}\left(\frac{L_1^2 + r^2 - L_2^2}{2L_1r}\right)$$

$$\theta_2 = \pi \pm cos^{-1}\left(\frac{L_1^2 + L_2^2 - r^2}{2L_1L_2}\right)$$

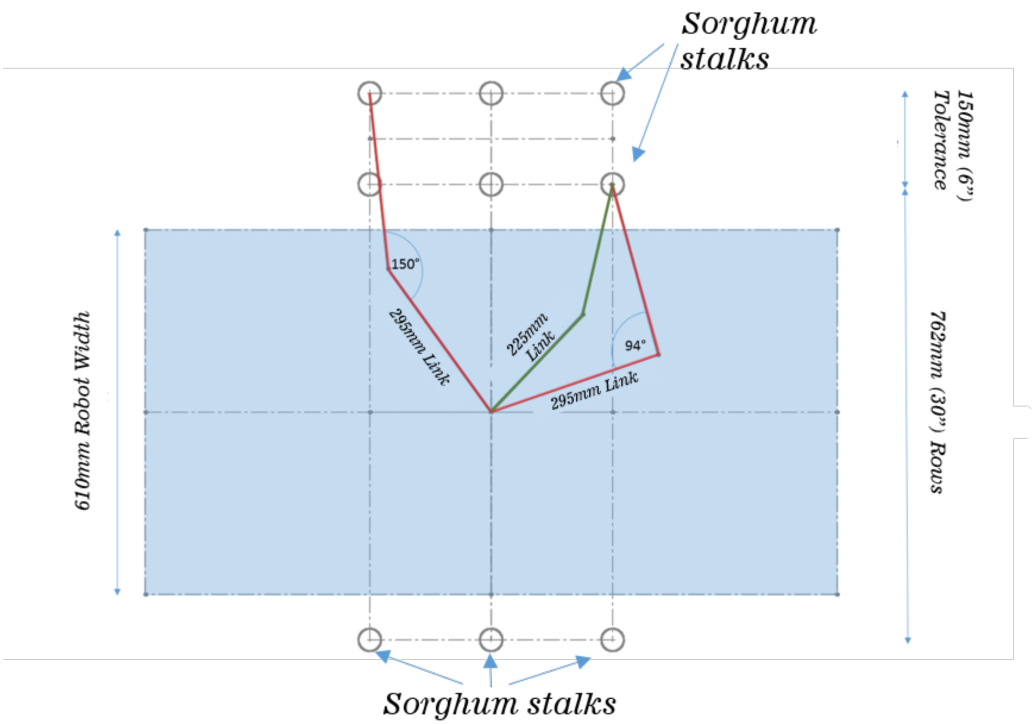


Figure 4.3: Arm length requirements for stalk grasping.

Calculating the inverse kinematics online is extremely fast, and takes approximately 60 microseconds in C++ using an i7 processor.

Arm lengths were determined by evaluating the geometric constraints imposed by sorghum plots. It is critical that a manipulator be able to reach all plants from the center of the row, so the arm length is a function of the maximum reach perpendicular to a row. In order to avoid singularity constraints, the arm was designed to never require full extension when grasping stalks. The length of the individual links is a function of the size of the end-effector, which is described in section 4.5.

4.3 Motor Selection and Sizing

The process of sourcing motor joints was similar to evaluating off-the-shelf manipulators based on requirements such as nominal torque, nominal speed, cost, and IP rating. Several companies fabricate motor modules that include a motor, gearbox, and controller in a single unit, and these are outlined in Table 4.4.

Table 4.4: Motor Module Performance Specifications

Manufacturer	Model	Mass (kg)	Cont. Torque (Nm)	No-load Speed (rpm)	Cost	Cont. Power Draw (W)	IP Class	Backlash (arcmin)
Kinova	K-58	0.357	3.6	15	$2,500	Unspecified	X2	7
Kinova	K-75	0.587	9.2	7	$4,000	Unspecified	X2	5
Kinova	K-75+	0.57	12	9.5	$5,200	Unspecified	X2	5
HEBI	X5-1	0.315	1.3	90	$1,750	20	65*	15
HEBI	X5-4	0.335	4	32	$1,750	20	65*	15
HEBI	X5-9	0.36	9	14	$1,750	20	65*	15
Schunk	PRL +80		21	4.5	$8000	120	54	Unspecified
Robotis	H54-200-S500-R	0.91	44.2	33.1	$2,590	150	Unspecified	3.5
Robotis	L54-30-S500-R	0.591	5.4	23	$890	40	Unspecified	5
Robotis	L54-50-S500-R	0.656	14	16	$990	50	Unspecified	5

*except for ethernet port

The motor modules manufactured by HEBI robotics offer several advantages: a low-profile rotary joint, Ethernet communication, and series-elastic actuation. The geometry of the HEBI motor modules fit well in the planar RR configuration because the module is thinnest along the axis of rotation. The vast majority of communication on the robot base is performed over Ethernet, so this communication architecture fit well in the existing system. Lastly, the series elastic actuator in the HEBI module offers both an additional layer of safety and exceptional torque feedback. The torque feedback has not been utilized, but it would be an elegant way of detecting collisions in the future.

The HEBI motor modules come in three different gear ratios. Each module was evaluated based on the time required to make a 90° joint movement and the percentage of time spent within the continuous operation range of the torque/speed curve. The moment of inertia of the individual arm links were modeled as a point mass and a rod, and modeling of the motor performance was implemented in Matlab. The results are outlined in Appendix A.

Other necessary analyses include shock loads and fatigue life of the motor bearings, shock loading on the manipulator joints, and bending moments on the vertical prismatic joint. These are outlined in Appendices B and C. The selected prismatic joint is a Movopart M75 fabricated by Thomson Linear Motion. This linear stage features a magnetic cover to prevent dust and water ingress, a slide guide that is robust to humidity and temperature changes, and a belt drive for rapid movement and light weight. The specifications of the linear stage are outlined in Appendix D.

4.4 Manipulator Control

Manipulator control is greatly simplified by the limited degrees of freedom of the arm and the semi-structured planting of sorghum rows. Although trajectory planning such as A* may provide the ability to reach around other plants, the advantages of of a more complex trajectory are constrained by limited degrees of freedom. A linear trajectory between the second arm joint and the target will minimize the area swept by the arm. Although this simple trajectory will not be able to avoid occluding objects within the travel path, it will be more robust to objects that are outside of the sensor's field of view. Computation of the optimal trajectory takes approximately 12 microseconds, and merely consists of parameterizing a 2D line in the arm plane between the first joint and the target.

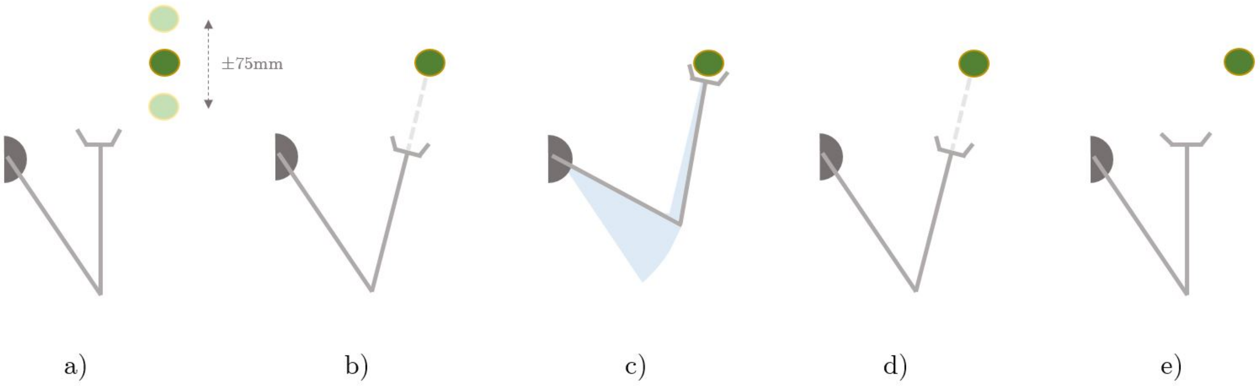


Figure 4.4: Manipulator trajectory and gripping routine: a) The manipulator begins in a fixed location while image processing is performed; b) The second link of the manipulator is positioned directly in line with the target; c) The end-effector takes a linear path to the target, disturbing minimal area; d) The end-effector follows a linear path away from the target; e) The manipulator returns to its starting position.

The arm trajectory is controlled using PID on position error, and joint velocity is controlled proportional to position error. Error estimation takes place at approximately 250Hz and the arm typically takes 2s to travel to a target.

The motor actuating the linear stage is controlled with a PI-controller. This is because the moment of inertia of the carriage on the linear stage is very small and a derivative term is not necessary to prevent overshoot. All three motors were tuned manually using the quarter amplitude decay method.

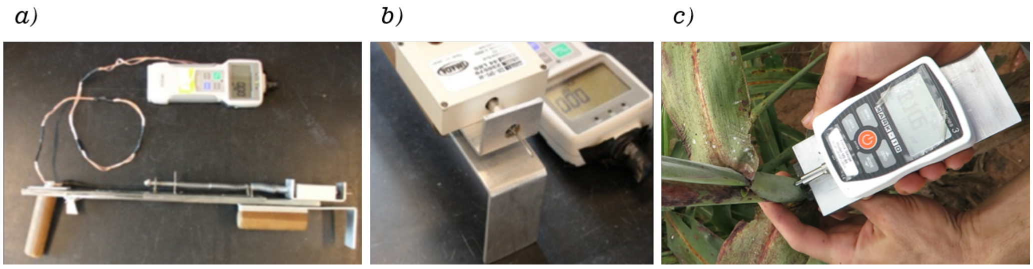


Figure 4.5: Handheld penetrometer designs for stalk strength measurements: (a) spring-loaded penetrometer [48]; (b) close-up of load cell attached to spring-loaded mechanism [48]; (c) penetrometer developed by the author for ground-truth testing.

4.5 End Effector Design

The sole purpose of the end-effector is to deploy a rind penetrometer. The end-effector must accommodate different stalk diameters and strengths as well as be robust to manipulator positioning inaccuracies. The exact requirements are outlined in Table 4.5.

Table 4.5: End-Effector Requirements

Mass	Less than 2 kgs
Sealing	Mechanisms concealed from the environment
Penetrometer Accuracy	Measure force up to 200N with 0.25N resolution
Travel Length	80mm minimum travel
Speed	Full travel in less than 1s
Travel Accuracy	Measure stalk diameters with 1mm accuracy
Torque	Apply force greater than 200N
Design	Mechanically simple and easy to disassemble

The process of autonomously applying a rind penetrometer requires two operations: 1) positioning a support finger behind the stalk, and 2) pushing the needle into the stalk. Several mechanisms were investigated for these two operations.

A system that plunges a needle into a stalk requires linear motion, and traditional methods of converting motor torque into linear motion include the following: Acme screw, ball screw, and rack and pinion. Unfortunately, there currently do not exist off-the-shelf linear actuators that meet the combined requirements of travel length, torque output, and speed, and a rack and pinion would not easily fit within the desired arm link geometry. Maxon Motor Corporation fabricates ball screws and Acme screws that are integrated with motor gearboxes, and ball screws offer very high efficiency in a small package. The final design for the plunger mechanism consists of a Maxon EC16 BLDC motor and a Maxon GP16S stainless steel ball screw.

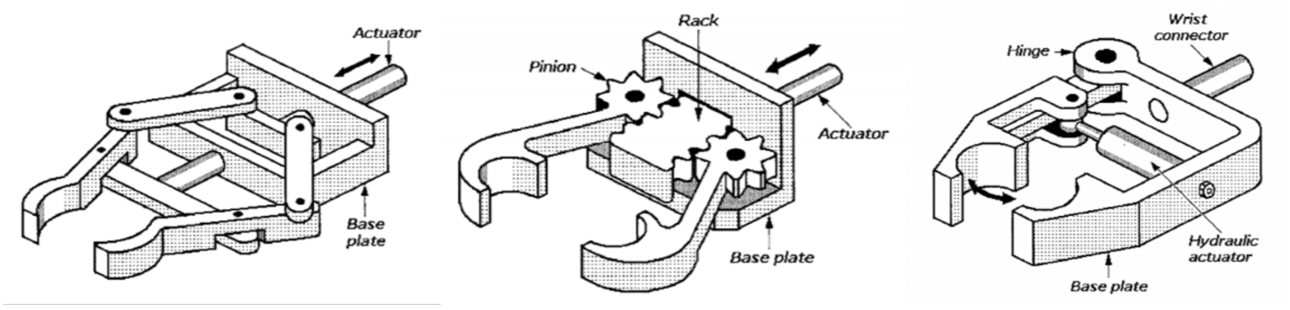


Figure 4.6: Mechanical gripping devices evaluated for stalk grasping [49].

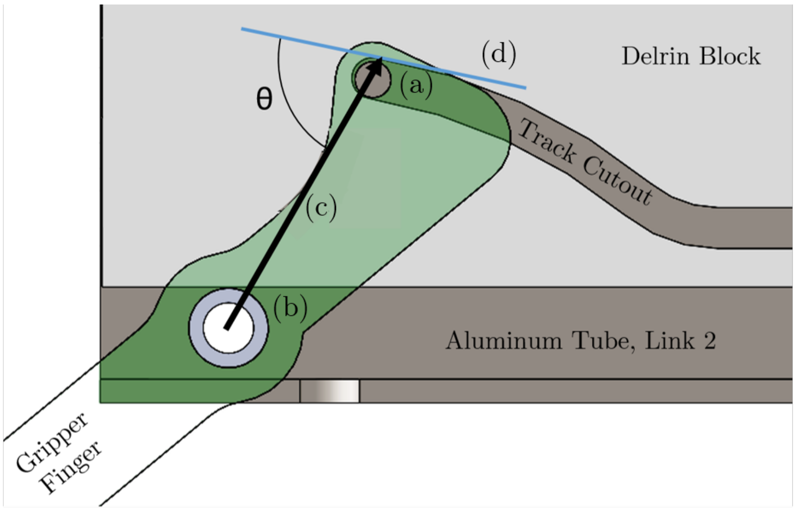
The secondary mechanical operation requires a finger or multiple fingers to be positioned behind the stalk to support it as the needle penetrates. Several designs are depicted in Figure 4.6. The mechanisms and their evaluation criteria are outlined in Table 4.6.

Table 4.6: Table of Actuation Mechanisms

Mechanism	**Evaluation Criteria**			
	Concealed	Simple	Compact	Robust
Pair of RC servos		X		
Single RC servo coupled to linkage	X	X	X	
Worm gear	X			X
Rack and pinion	X			X
Coupled sliding mechanism	X	X	X	X

A coupled sliding mechanism offers the advantages of extreme simplicity, compact geometry, and full concealment. Delrin was selected as the optimal material for a sliding mechanism due to its wear resistance and very low coefficient of friction on steel. Figure 4.7 shows the estimated slip point between Delrin and a steel pin in the coupled sliding mechanism, which was calculated at 78.5°. Empirical testing was performed with hand-actuated prototypes to confirm this slip point, and the true slip point was determined to be approximately 70°.

A v-block was designed to center the stalk relative to the end-effector before the needle touches the wall of the stalk. Four pins allow the v-block to slide toward the main Delrin block as the gripper closes around a stalk. This both conceals the needle until contact with the stalk and provides a method of determining stalk diameter. Successful closure of the gripper around a stalk pushes the v-block back toward the arm link, which completes a circuit. Diameter estimation is accomplished by reading



$$Friction : \mu \cdot F \cdot sin(\theta)$$

$$Slidingforce : F \cdot cos(\theta)$$

$$\mu \approx 0.2$$

$$\mu \cdot F \cdot sin(\theta) = F \cdot cos()$$

$$cot(\theta) = \mu$$

$$\theta = 78.5^\circ (calculated)$$

$$\theta = 70^\circ (empirical)$$

Figure 4.7: Visualization of the slip point between (a) a steel pin mated to the gripper finger and a slot in the Delrin plunger. The steel pin is attached to the rotational joint (b) which applies a force (c) to the Delrin block (d). The image shows a section view of the front portion of the final gripper design; full detail can be seen in Figure 4.8.

the encoder position of the actuation motor when the v-block circuit closes.

The constrained packaging of the gripper necessitated a very small load cell. The load cell that best fit within the design constraints was a button-style load cell manufactured by Futek, Inc. that has a diameter of 9.5mm. The LLB215 model registers a maximum force of 222N (50 lbf) and has a non-repeatability of 0.1% of rated output. An instrumentation amplifier manufactured by LabJack Inc. was mounted inside the sliding Delrin block to minimize transmission length and noise on the differential signal. The final gripper design is shown in Figure 4.8. The process of actuating the Delrin plunger block forward and backward causes the gripper fingers to open and close. The gripper fingers have a fixed pivot point but slide on the Delrin block.

4.6 Field Testing

Testing and validation of the automated penetrometer were performed in Clemson, SC. An exceptionally dry summer caused many of the sorghum stalks to be too thin for the penetrometer measurement, so testing was performed on 46 hand-selected stalks greater than 20mm in diameter. Each stalk was punctured once with the automated penetrometer and six times with the manual penetrometer. The

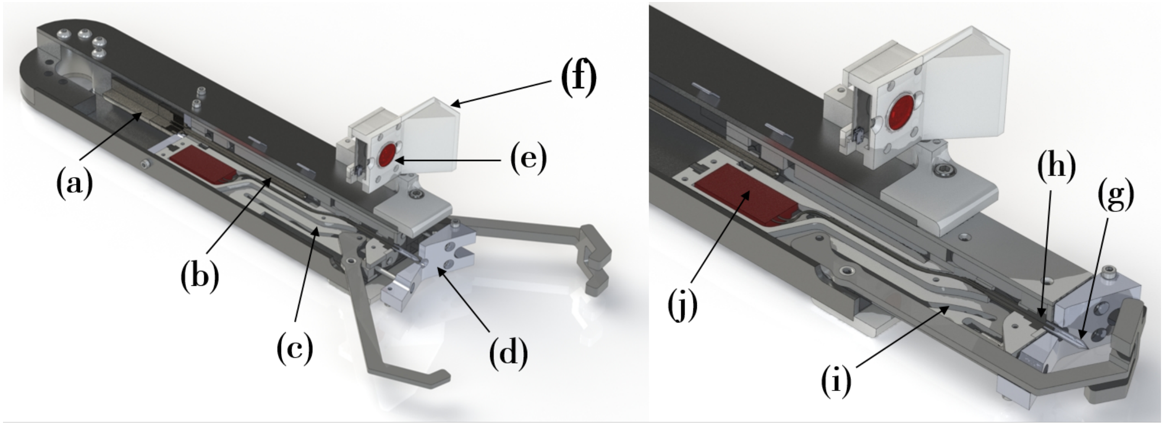


Figure 4.8: Cutaway of the custom gripper: (a) Maxon Motor EC16 brushless DC motor; (b) Maxon Motor GP16S ball screw; (c) acetal plunger block; (d) v-block for centering plants; (e) Code Laboratories DUO3d stereo camera; (f) sun shield; (g) needle for penetrating stalk walls; (h) Futek llb215 load cell; (i) track for actuating fingers; (j) instrumentation amplifier and A2D converter.

mean of the manual penetrometer measurements for each stalk were treated as ground truth. Figure 4.9 shows a histogram of the number of measurements that fell above or below the mean.

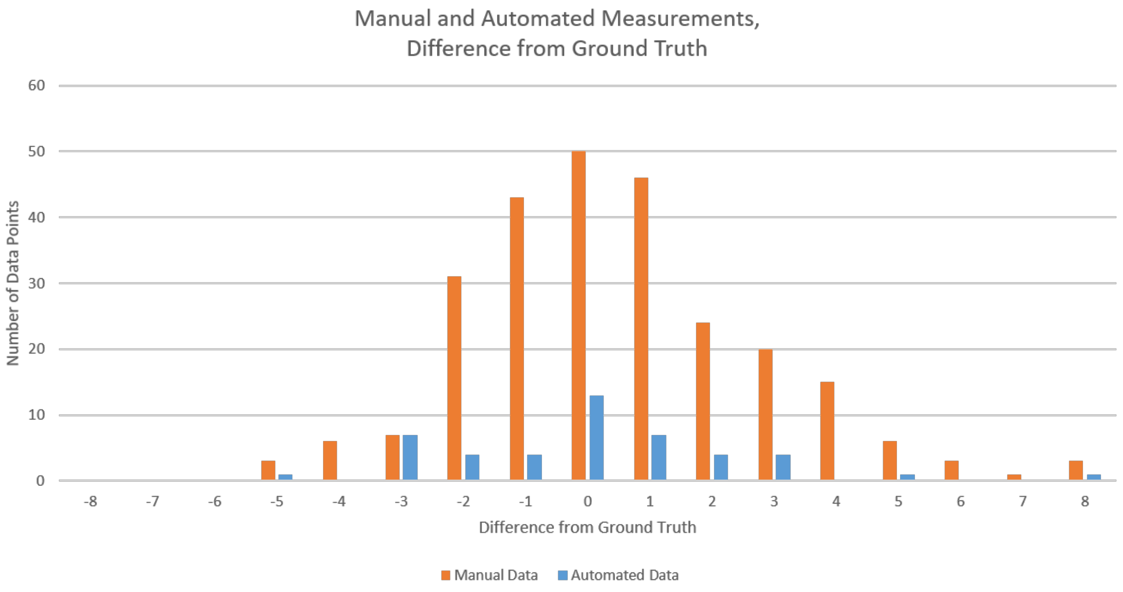


Figure 4.9: Histogram of penetrometer measurements and their difference from ground truth.

A two one-sided equivalency test shows the degree to which the two sets of measurements belong to the same dataset. As the name implies, this test consists of two separate equivalency tests, each with a null hypothesis. If the null hypothesis can be rejected from both tests, then one can conclude within a confidence level that the two datasets come from the same distribution. The data from the penetrometer test allows us to reject with 99% confidence the null hypothesis that population means differ more than ± 15% from the average force reading. The full analysis is detailed in Appendix E.

Chapter 5

Stalk Detection: RANSAC Cylinder

5.1 Setup

This chapter describes stalk detection using brute force geometric methods. The sensor used for stalk detection was a DUO3D MLX, which outputs two 0.3 MP grayscale images at approximately 10 Hz. Correspondence matching consisted of generic block matching using the *stereo_image_proc* package in ROS. Stereo processing consumed nearly 100% of CPU processing capabilities (3.4 GHz Intel i7 processor, 16 GB RAM).

The camera was mounted on the end of the manipulator in a housing designed to provide protection from dust ingress as well as shading from excess sunlight. The camera cover was also fitted with polarizing lenses to reduce glare and overexposure. The camera, housing, and lenses are visible in Figure 5.1.

5.2 Pipeline

Point cloud processing was performed with Point Cloud Library (PCL) and communication was organized with ROS. After the point cloud was converted to PCL format, dark pixels were removed from the point cloud. These dark pixels are associated with the background and incorrect correspondence

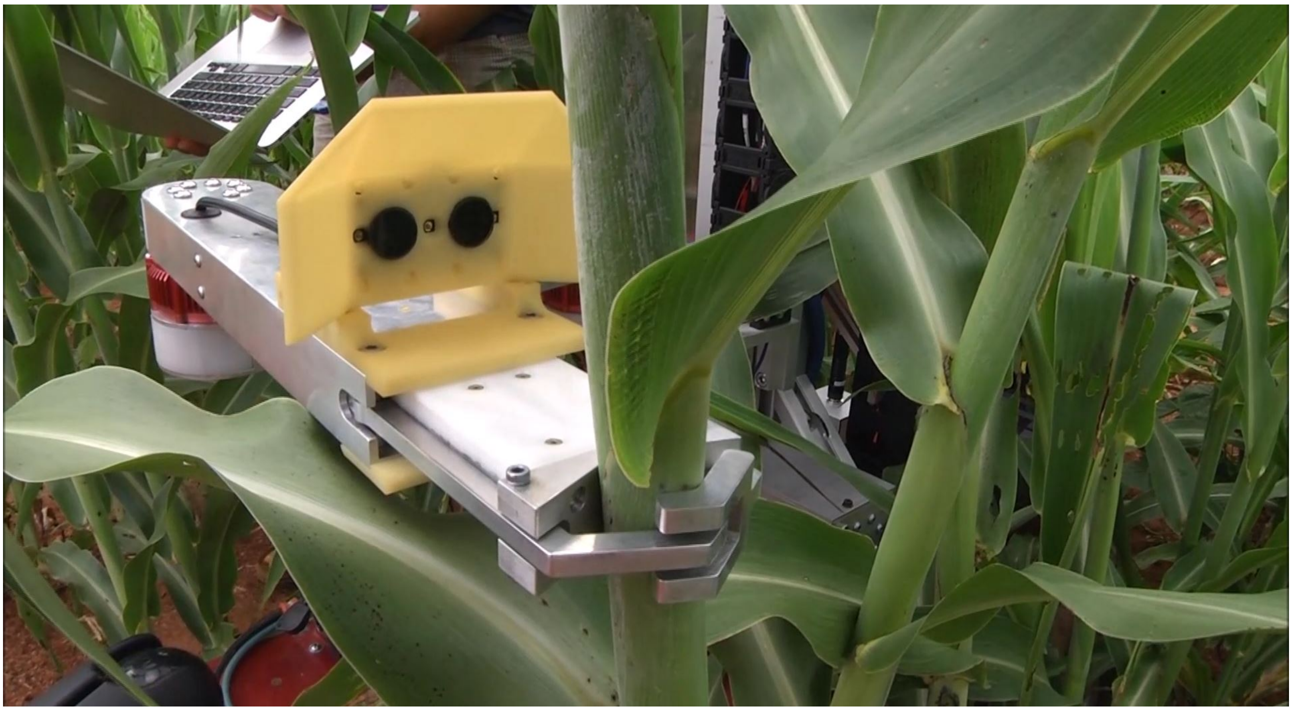


Figure 5.1: The DUO3D stereo camera mounted on the end of the manipulator.

matches. The cloud was then filtered along the camera's z-axis to remove noise in the foreground and background.

Cylinder detection was performed using RANdom SAmple Consensus (RANSAC). There are several ways of defining a cylinder, including the two-point method, the five-point method, and the seven-point method, but the two-point method requires the fewest points during each iteration. Although the two-point estimation method requires the calculation of point normals, the smaller number of correspondence points provides enormous computational advantages. The number of iterations k required to select correct inliers can be defined as a function of several variables [50]:

$$w = \frac{\text{number of inliers in data}}{\text{number of points in data}}$$

$$n = \text{points needed to estimate a model}$$

$$p = \text{probability that a model is correctly matched to data}$$

$$k = \frac{log(1-p)}{log(1-w^n)}$$

As the number of points parameterizing a cylinder increases, the number of iterations required to match a model to data with probability p increases to the power of the number of model parameters as shown in Figure 5.2.

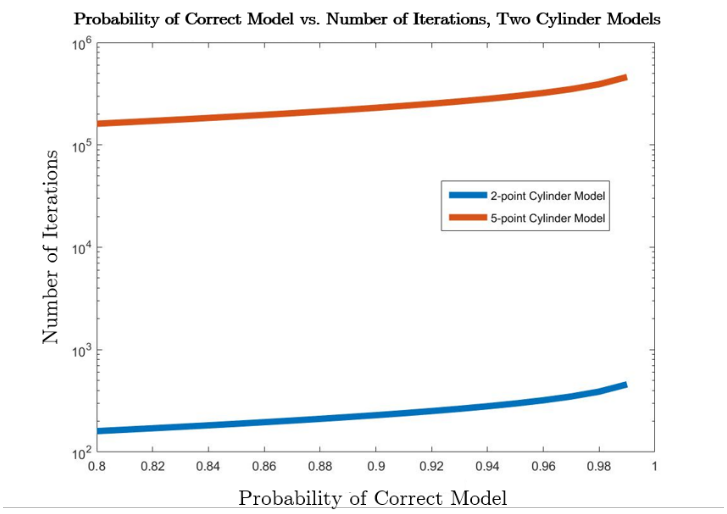


Figure 5.2: The number of iterations required to successfully detect a geometric primitive using RANSAC is a function of the number of points parameterizing the primitive.

Applying constraints to cylinder detection ensures that the model fulfills defining aspects of a stalk, such as radius, lean angle. The parameters and their values are described in Table 5.1. The pseudocode for RANSAC cylinder detection applied to sorghum stalks is detailed in Algorithm 1.

Table 5.1: Table of variables and values for RANSAC cylinder detection

Variable	Value	Description
$N_{iterations}$	10,000	Number of iterations before completing RANSAC
θ	0.17 rad	Maximum angle from vertical
r_{max}	20mm	Minimum cylinder radius
r_{min}	30mm	Maximum cylinder radius
r_{tol}	10mm	Distance tolerance on points
r_{2D}	5mm	Search radius for 2D inliers after performing RANSAC

Although RANSAC is effective at fitting cylinders to point clouds, cylinders are occasionally fit to vertical leaves rather than stalks. One way of overcoming this issue is to run RANSAC multiple times. Variations in lighting, correspondence matching, and wind generate sightly different point clouds at each iteration. This variation, combined with the random point selection of RANSAC, is enough to detect cylinders in different locations of the point cloud. Subjective analysis indicates that, across multiple RANSAC iterations, stalks are detected with a higher probability than leaves. By running multiple iterations of the RANSAC algorithm and detecting clusters of detected cylinders, the likelihood of detecting a leaf as a cylinder is reduced.

In implementation, cylinder detection is repeated 30 times from a stationary manipulator. The intersection of each cylinder's axis and the manipulation plane is plotted in 2D on the manipulator plane. The number of neighbors within radius r_{2D} is calculated for each intersection point, and the cylinder with the most neighbors is deemed the optimal stalk location. The manipulator is then commanded to the optimal stalk location. A visualization of this approach is shown in Figure 5.3. This processing framework takes approximately 10 seconds.

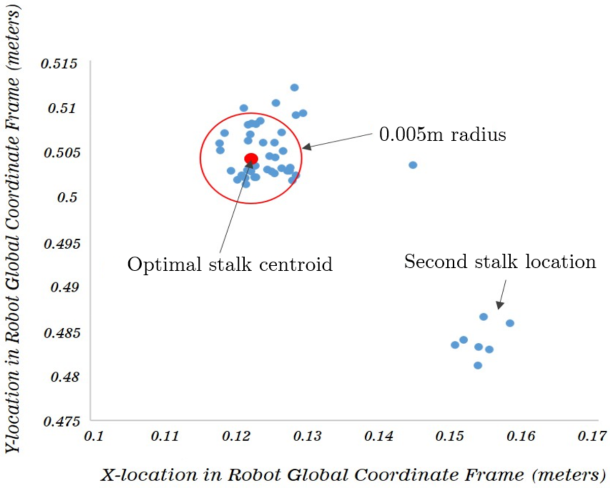


Figure 5.3: Thirty stalk locations are estimated from field data and plotted on a 2D graph. Two stalks are detected in this image, and the optimal target is determined by the most neighbors within a 0.005m radius.

5.3 Field Testing

Field testing was performed in two different locations in South Carolina. In August, 2016, the RANSAC detection method was tested on a commercial variety of energy sorghum in Florence, SC. This particular commercial variety of energy sorghum exhibited ideal conditions for detection and grasping: large stalks (>25mm in diameter) and minimal tillering (single stalks extending from the a root base). A wooden base platform supporting the manipulator and manipulator-mounted stereo camera was dragged through approximately 7.5m of sorghum. Out of 26 stalks imaged, 25

Algorithm 1 Stalk Detection Using Two-Point Cylinder RANSAC

INPUTS:

Point cloud: cloud, point cloud normals: cloud_normals

OUTPUTS:

Single stalk location parameterized by a cylinder: cyl_params

FUNCTIONS:

bool angleCheck(cyl_params, θ)
▷ checks that the angle of the cylinder axis is within tolerance θ

bool radCheck(cyl_params, r_{min}, r_{max})
▷ checks that the radius of the cylinder is between r_{min} and r_{max}

int inliersCount(cyl_params, pt_cloud, $\mathrm{r_{tol}}$)
▷ counts the number of points that lie within distance $\mathrm{r_{tol}}$ of the cylinder radius

```
prev_inliers ← ∅                                            ▷ integer for saving the inlier count
for $i < N_{iterations}$ do
    rand_ints = rand([0, cloud.size()], 2)                  ▷ select two random points from the cloud
    $p_1$ = cloud[rand_ints(0)]                              ▷ vectors assigned to points
    $p_2$ = cloud[rand_ints(1)]
    $n_1$ = cloud_normals[rand_ints(0)]                      ▷ vectors assigned to normals
    $n_2$ = cloud_normals[rand_ints(1)]
    axis = $n_1$.cross($n_2$)                                ▷ cylinder axis direction
    temp = $n_2$.cross($n_1$.cross($n_2$))
    cyl_center = $p_1 + \frac{(p_2 - p_1).\mathrm{dot(temp)}}{n_1.\mathrm{dot(temp)}} n_1$    ▷ location of shortest dist. between two normals
    radius = $\|\mathrm{axis} - p_1\|$                       ▷ calculate radius
    cyl_params = [axis, $p_1$, radius]                       ▷ axis vector, point on axis, radius
    if angleCheck(cyl_params, $\theta$) then                 ▷ check cylinder angle
        if radCheck(cyl_params, $\mathrm{r_{min}}$, $\mathrm{r_{max}}$) then    ▷ check cylinder radius
            num_inliers = inliersCount(cyl_params, pt_cloud, $\mathrm{r_{tol}}$)    ▷ count number of inliers
            if num_inliers > prev_inliers then
                save(cyl_params)                             ▷ save cylinder parameters
                prev_inliers = num_inliers
```

were successfully detected and grasped. The cause of the one unsuccessful detection was a curled vertically-oriented leaf.

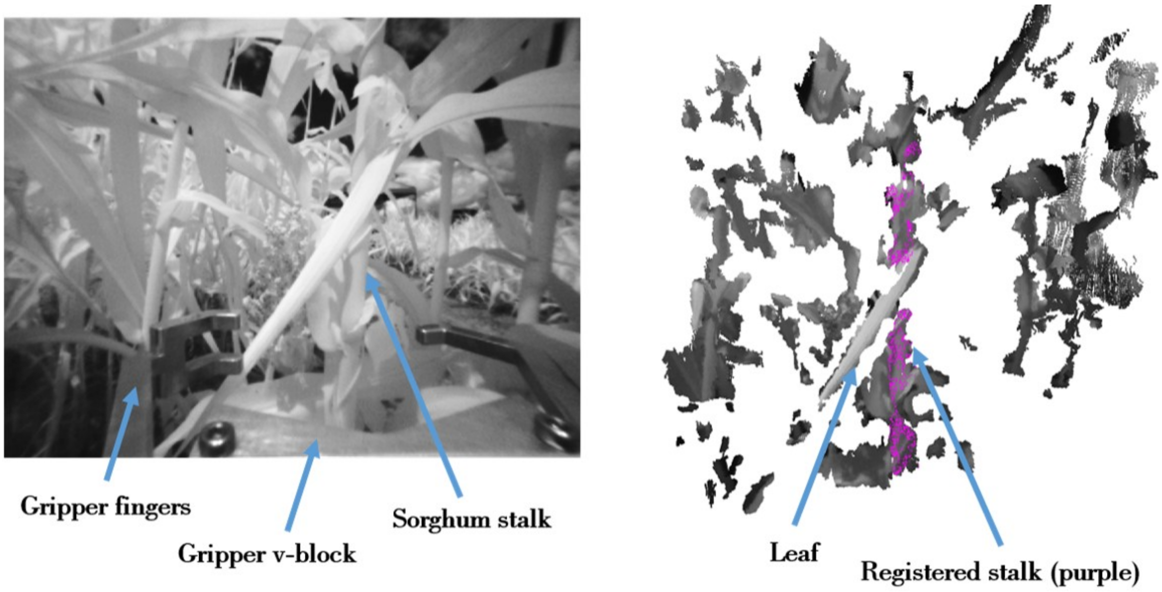


Figure 5.4: Field data from the DUO3d stereo camera: (Left) 2D grayscale data from the left camera - the stalk is highly occluded and a vine is wrapped around it; (Right) the geometry-based stalk detection algorithm successfully registers the stalk.

A second round of testing was performed in September, 2016 in Clemson, SC. An image of successful stalk detection can be seen in Figure 5.4. The manipulator was mounted on the robot base and the robot was teleoperated down a 60-meter row of sorghum. 70 out of 75 stalks were successfully detected, and 64 out of 70 stalks were successfully grasped. The cause of unsuccessful detections were vertical leaves, and the unsuccessful grasps were due to thin vines obstructing a clear path to the target stalk.

Chapter 6

Stalk Detection: Point Cloud Stitching

Tests outlined in Chapter 5 indicate that brute-force cylinder detection identifies stalks with >90% accuracy but requires significant computation time and only detects one stalk at a time. The detection method outlined in Chapter 5 was developed under two constraints: lack of color information and poor stereo reconstruction. This chapter investigates stalk detection without those constraints. Processing and analysis are performed offline but with the intention of applying the algorithm to in-field detection.

6.1 Pipeline

6.1.1 Data Collection

Images of *Sorghum bicolor* were captured in July, 2016 in Pendleton, South Carolina. The imaging sensor consisted of a custom 9MP stereo camera synchronized to Xenon flashes as described in [51]. Active lighting dramatically improves disparity matching by increasing contrast between foreground and background. The stereo camera was mounted to a UGV at a height of approximately 150mm above ground and positioned approximately 500mm from the plants. The camera was operated over USB3 by an Intel NUC Mini-PC. The robot was teleoperated through rows of sorghum at a speed of approximately 0.5 m/s, and the camera was triggered at a rate of 5Hz. Image pairs were saved to memory through Robot Operating System (ROS).

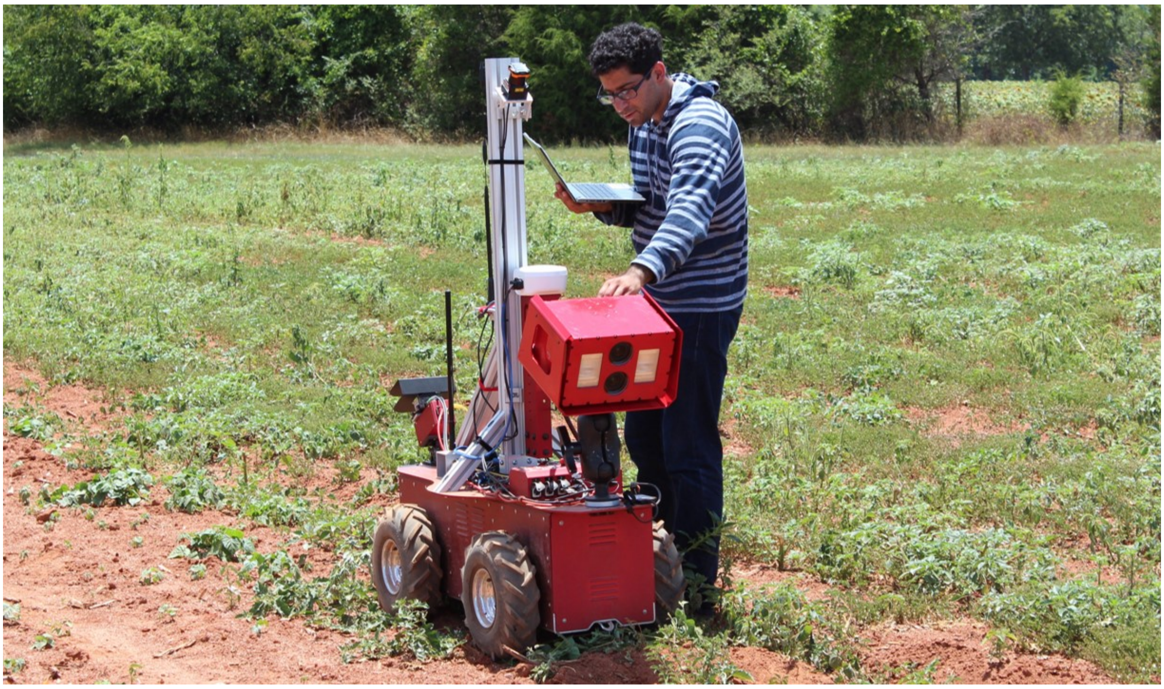


Figure 6.1: Stereo camera mounted on “The Robotanist” [45].

6.1.2 Stereo Matching and Point Cloud Stitching

Each image pair is converted to a 3D point cloud by downsampling images 4X and then performing semi-global block matching (SGBM) in OpenCV. The point clouds are then converted into the native format of Point Cloud Library (PCL) and stored in a KD-tree. The individual point clouds are filtered to remove extraneous points. Pixels with RGB intensity values below (15,15,15) are treated as incorrect correspondences and removed. A statistical outlier filter then calculates the mean distance between each point and its K nearest neighbors, removing points associated with distances greater than one standard deviation from the mean [52].

In order to minimize the effects of visual occlusion from any single image, five point clouds are stitched together. Stitching is achieved using linear iterative closest point (ICP), which is implemented through PCL [52]. The maximum correspondence distance between points was set to 0.15m, and the cloud was downsampled using a 3cm x 3cm x 3cm voxel grid for faster correspondence search. The

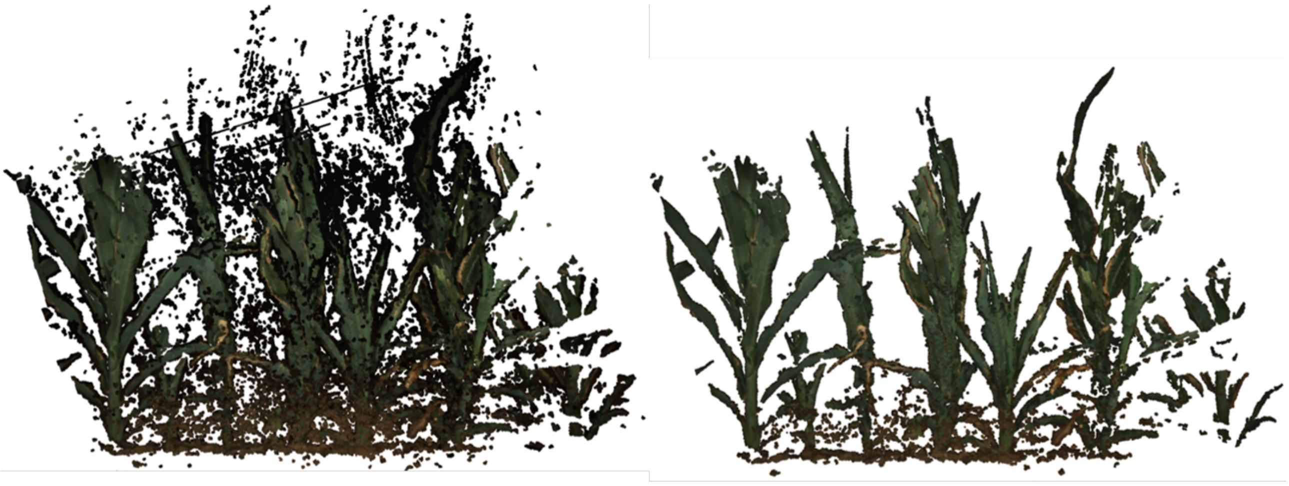


Figure 6.2: Left: point cloud before filtering; Right: point cloud after removal of dark pixels and statistical outliers.

average time required to stitch each cloud and process other elements of the pipeline is captured in Figure 6.6.

During image capture, the ground was not always parallel to the horizontal axis of the stereo camera. Approximately 70% of the brown pixels in each image correspond to the ground plane, and the rest correspond to necroded leaves, so RANSAC is a reliable method of extracting the ground plane. The cloud is then rotated so that the ground plane is horizontal, which provides a global reference for stalk extraction.

6.1.3 Stalk Extraction

The proposed algorithm relies on the vertical continuity and erectness of stalks. This is captured in four defining geometric properties:

- A stalk will exhibit a high point density along its axis.
- Patch-based normals on the surface of a stalk will be roughly parallel to the ground plane
- A stalk will be continuous throughout a significant portion of an image
- A stalk will be visible close to the ground plane

The high relative point density of stalks is captured by slicing the point cloud into 0.2m segments as shown in Figure 6.3 and collapsing the points to a 2-D plane described by the x-y axis. This plane, when sectioned into 1cm x 1cm bins, represents a 2D histogram of point density.

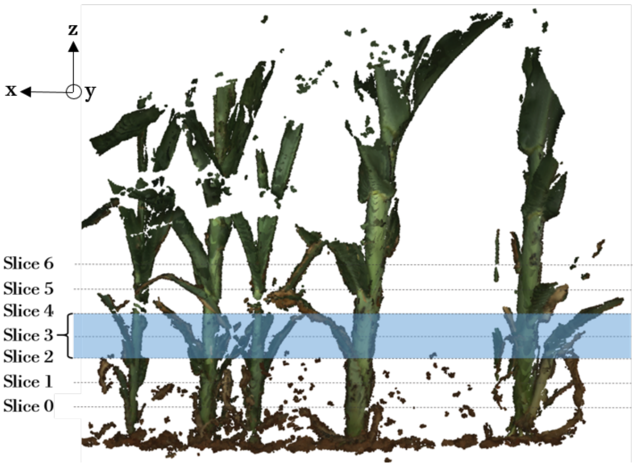


Figure 6.3: The point cloud is sliced into overlapping layers 0.2m in height.

Patch-based normal estimation is locally noisy but globally descriptive. Each bin of the 2D histogram is weighted according to the mean normal angle of the points captured in that bin. The value of the bin after weighting is:

$$\text{bin}_{\text{value}} = \text{bin}_{\text{value}} * (1 - \text{abs}(\cos(\phi)))$$

The variable ϕ is the angle of the average patch normal associated with that bin, and $1 - \text{abs}(\cos(\phi))$ is the z-component of ϕ. Stalk point normals should exhibit a very small z-component. The corrected 2D histogram is then thresholded by a minimum value α in order to minimize noise. The result is depicted in Figure 6.4.

In order to extract stalk location information from the corrected histogram, a simple 8-connected neighborhood region growing algorithm segments the histogram into patches. The region growing algorithm is initialized by seed points, which are defined as bins representing more than β points per cm^2, where β is a user-defined parameter. The result is visualized in Figure 6.4, which also clearly shows two centroids placed on leaves. Point density and patch-based normal estimation are imperfect classifiers but good initializers for 3D methods.

After 2D projection and region growing is applied to every slice, the centroids of each region are reprojected to 3D. This is performed by recording the x-y location of each centroid in the histogram and positioning the centroid at the z-location of its associated slice. These centroids are visible as

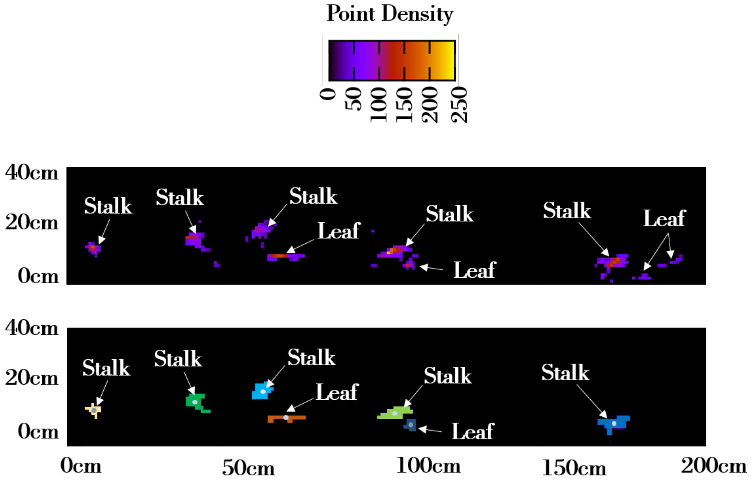


Figure 6.4: Top) The third slice of the point cloud depicted in Figure 6.3 is projected onto the x-y plane. Stalk density is plotted in 1cm x 1cm bins as a heat map, bin weights are corrected based on patch normals, and low-frequency noise is removed. Bottom) The region-growing algorithm segments high-density regions and calculates the centroid (gray).

yellow spheres in Figure 6.5. For each centroid located in Slice 0 or Slice 1 (i.e. centroids close to the ground), a cone-shaped upward search region is parameterized by angle θ. This cone-shaped search region is depicted in Figure 6.5.

A variation of the RANSAC algorithm is applied to centroids that fall within the conical search region. A line is defined between the base centroid and each centroid in its search space. For each line, the number of centroid inliers is recorded, parameterized by an inlier threshold ρ. The optimal connecting line is selected based on the maximum number of inliers. If two lines contain the same number of inliers, the tie is broken by measuring the sum of squared distances between the line and each inlier. To avoid detecting vertical leaves, a minimum number of centroids must be detected along the line, parameterized by $\text{inliers}_{\text{min}}$. This simple method is O(n), where n represents the number of centroids in a search space. The results of this search method are depicted in Figure 6.5.

Table 6.1: Stalk Detection Variables and Values

Variable	Value	Description
N_{slices}	6	Number of slices
α	40	Histogram noise threshold
β	80	Histogram seed point threshold
θ	0.2 rad	Search cone angle from vertical
ρ	0.025m	Inlier cutoff from RANSAC line
$\text{inliers}_{\text{min}}$	4	Minimum inliers defining a stalk line
step	0.1m	Height increment of slices

The parameter values tested in the described algorithm are listed in Table 6.1. Pseudocode outlining the stalk extraction algorithm is described in Algorithm 2.

Algorithm 2 Stalk Detection

INPUTS:

Stitched point cloud

OUTPUTS:

Stalk locations parameterized by line segments or cylinders

FUNCTIONS:

$\mathrm{regionGrow(seed, histogram)}$

▷ applies 8-connected neighborhood region-growing to the 2D histogram starting at seed locations

$\mathrm{lineRANSAC}(S, \mathrm{inliers_{min}}, C)$

▷ finds a line with the most inliers between ground centroid C and cone-bounded centroids S; checks that number of inliers is greater than or equal to $\mathrm{inliers_{min}}$

```
centroids ← ∅                                          ▷ vector for region-grown centroids
for $i < N_{slices}$ do
    Project $i^{th}$ point cloud slice to x-y plane
    Create 2D histogram of point density
    De-weight histogram bins based on patch normals
    seed ← ∅                                           ▷ vector for region-growing seeds
    for bin in histogram do
        if $\mathrm{bin_{value}} < \alpha$ then
            $\mathrm{bin_{value}} \leftarrow 0$
        if $\mathrm{bin_{value}} > \beta$ then
            $\mathrm{seed.append(bin_{location})}$
    $\mathrm{centroid}s[i] = \mathrm{regionGrow(seed, histogram)}$
Reproject centroids to 3D                              ▷ location $c_x$, $c_y$, $c_z$
cylinder ← ∅                                           ▷ vector for stalk locations
for $C$ in $\mathrm{centroids}[0, 1]$ do                ▷ centroids near ground
    $S \leftarrow \emptyset$                           ▷ vector for centroids
    for $c$ in centroids do                            ▷ all centroids
        $\mathrm{bound} = (c_z - C_z) * \tan(\theta)$  ▷ x-y cone limits
        if $\sqrt{c_x^2 + c_y^2} < \mathrm{bound}$ then
            $S.\mathrm{append}(c)$                     ▷ centroids within cone
    $\mathrm{cylinder}[C] = \mathrm{lineRANSAC}(S, \mathrm{inliers_{min}}, C)$
Display stalks defined by cylinder
```

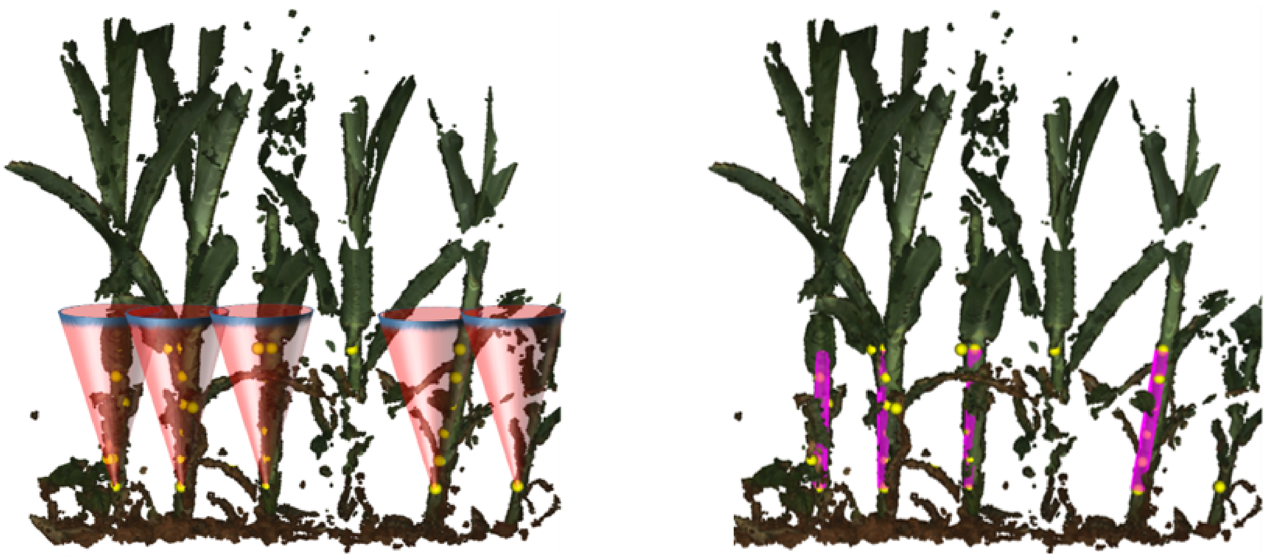


Figure 6.5: Top: potential stalk locations are depicted as yellow spheres, and connected points are located within an upward-facing cone; Bottom: optimal stalk angles calculated from RANSAC and depicted as magenta cylinders.

6.2 Algorithm Evaluation

The proposed stalk detection algorithm was tested on 378 stitched point clouds generated from 916 images. The point clouds were independent from the data used to develop the parameters outlined in Table 6.1. Images were captured at different times of day across different locations of a breeding plot, and represent approximately 110 meters of crop. Detection time took approximately 12s per stitched cloud.

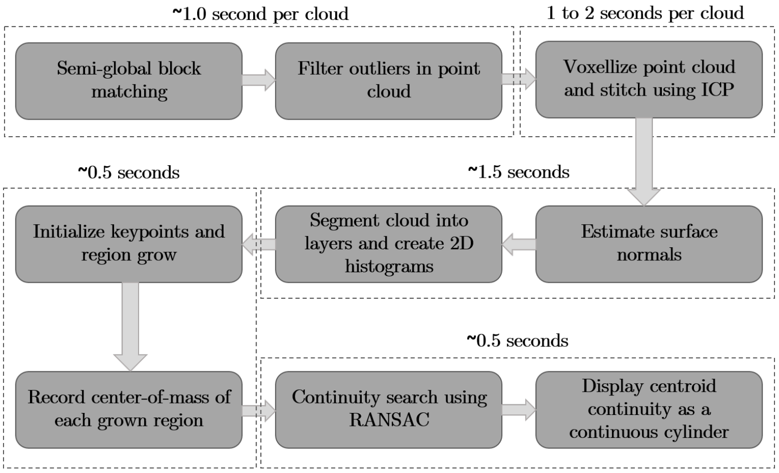

~1.0 second per cloud
1 to 2 seconds per cloud
Semi-global block matching
Filter outliers in point cloud
Voxellize point cloud and stitch using ICP
~0.5 seconds
~1.5 seconds
Initialize keypoints and region grow
Segment cloud into layers and create 2D histograms
Estimate surface normals
~0.5 seconds
Record center-of-mass of each grown region
Continuity search using RANSAC
Display centroid continuity as a continuous cylinder

Figure 6.6: Overview of the image processing pipeline.

The focus of algorithm evaluation is to determine the number of correctly classified stalks, incorrectly classified stalks, and total stalks present. Algorithm output was compared to ground truth (human labeled) point clouds. A cylinder is considered a true positive (TP) if its angle and position closely match that of the ground truth, and is considered a false positive (FP) otherwise. Any undetected stalk is considered a false negative (FN). An example of a false positive is depicted in Figure 6.7.

Precision is defined as the ratio between correctly classified stalks and the total number of positively classified stalks:

$$precision = \frac{TP}{TP + FP}$$

Recall is defined as the ratio between correctly classified stalks and all true stalks:

$$recall = \frac{TP}{TP + FN}$$

The F1 score is defined as the harmonic mean of precision and recall:

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

These three metrics are used to quantify the algorithm's effectiveness.

Consecutive processing of images accommodates two different approaches to algorithm evaluation: 1) classification accuracy on individual stitched point clouds, and 2) classification accuracy on a continuous stream of stitched point clouds. The two types of evaluation have different applications: stalk identification for grasping will almost exclusively be performed on a single stitched cloud, whereas stalk counting would be applied to a stream of clouds recorded as a robot drives through a field.

6.2.1 Discrete Stitched Point Cloud Evaluation

High precision is the primary identification goal for plant manipulation. It is unlikely that a robot will have to collect data from every plant, but there must be high confidence that it is grasping the intended object.

Out of 2641 ground truth stalks, the detection algorithm correctly classified 1358 stalks and incor-

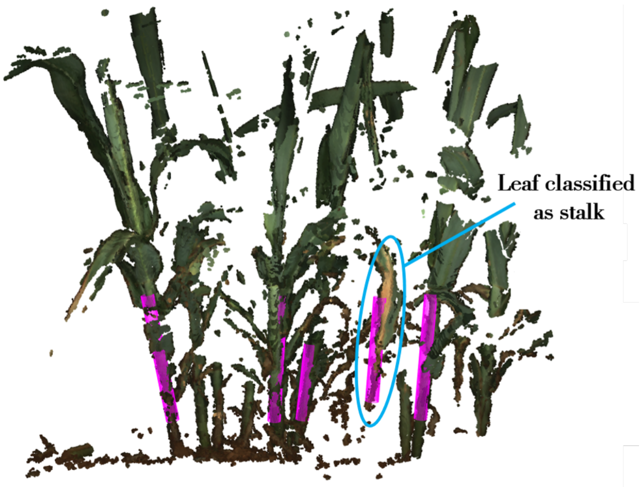


Figure 6.7: Example false positive stalk classification. Note that the leaf is both vertical and positioned close to the ground plane.

rectly classified 73 stalks. Classification results indicate a precision of 0.95, a recall of 0.51, and an F1 score of 0.67. Detection results are captured in Table 6.2.

Table 6.2: Table of Discrete Point Cloud Classification Results

		Predicted Condition	
		Positive	Negative
True Condition	Positive	1358	1283
	Negative	73	N/A

Precision	0.95
Recall	0.51
F1 Score	0.67

6.2.2 Continuous Stitched Point Cloud Evaluation

Stalk counting applications require not only high precision but also high recall. Both false positives and false negatives will lead to incorrect estimates.

The method applied in this evaluation involved tracking each stalk along the path of the camera rather than counting stalks in a single stitched point cloud. If the stalk was detected at any point along the camera's path, it was considered a positive detection.

Classification results indicate a precision of 0.93, a recall of 0.77, and an F1 score of 0.84. As is evidenced by higher recall in this evaluation method, applying the detection algorithm to the same stalk in different point clouds increases the likelihood that it will be detected. Detection results are captured in Table 6.3.

Table 6.3: Table of Continuous Point Cloud Classification Results

		Predicted Condition	
		Positive	Negative
True Condition	Positive	446	135
	Negative	31	N/A

Precision	0.93
Recall	0.77
F1 Score	0.84

6.3 Summary

The proposed algorithm has been demonstrated to be robust to occlusions and effective at detecting stalks with >93% precision. The detection pipeline takes approximately 20 seconds - a longer period of time than the algorithm outlined in Chapter 5, but with the capacity to detect an unlimited number of stalks.

Chapter 7

Stalk Detection: Grasp Point Detection

7.1 Motivation

The method outlined in Chapter 6 exhibits high precision on the test dataset, which exclusively consists of 2 month-old sorghum stalks imaged 150mm above the ground. Unfortunately the images in the test dataset are different than those that will be captured during a rind penetrometer test. This is because the rind penetrometer test is performed on older stalks, grasping is performed several tens of centimeters above the ground, and the imaging camera will be positioned closer to the plants due to robot architectural constraints. This introduces several factors that confound the algorithm described in Chapter 6:

- The ground plane will not be visible, so searching for stalk initialization points directly above the ground plane is infeasible
- Stalks will be more occluded due to both increased leaf density higher in the canopy and advanced plant maturity
- A short camera distance will reveal less of a stalk's height relative to leaves

This chapter describes an algorithm that employs methods similar to Chapter 6, but operates significantly faster and detects stalk grasp locations rather than parameterizing the entire stalk. This is necessary because images of highly occluded stalks may reveal only small stalk sections, which are insufficient for full parameterization but are sufficient for grasping.

Data collection was performed early in July 2017 in Clemson, SC on corn stalks. The sorghum growing season lasts from June through October, and the sorghum plants in early July were too small to be representative of fully-grown plants. A nearby corn field served as a substitute for mature sorghum, and differences between the plants are discussed in section 7.3.

7.2 Setup

A Multisense S7 stereo camera was mounted to the robot mast and synchronized to custom LED flashes visible in Figure 7.1. The camera was mounted at a height of 1m above the ground and 25mm in front of the robot's center plane. The camera mount positions the camera nominally 350mm from plant stalks. The minimum range of the S7 camera is approximately 200mm, so every imaged stalk lies within the minimum range of the S7 camera when it is mounted to the robot mast.

The S7 provides advantages over the DUO3D camera discussed in Chapter 5 and the custom imager discussed in Chapter 6. Unlike the DUO3D, the S7 captures color images and point clouds at resolutions up to 4 MP. Unlike the custom imager, the S7 is light and compact, and can be mounted the robot mast. Furthermore, it offers a smaller baseline (70mm) and minimum range, and an onboard field programmable gate array (FPGA) computes semi-global correspondence matching at rates up to 15 Hz.

The image capture routine consists of taking five images while autonomously driving forward at 0.5 m/s. Recorded data consists of 2D images, point clouds, wheel odometry, and manipulator joint states.

7.3 Pipeline

The focus of the pipeline is twofold: successfully detect stalk grasp points and optimize processing time. Processing time is reduced by a combination of hardware additions (wheel odometry and an

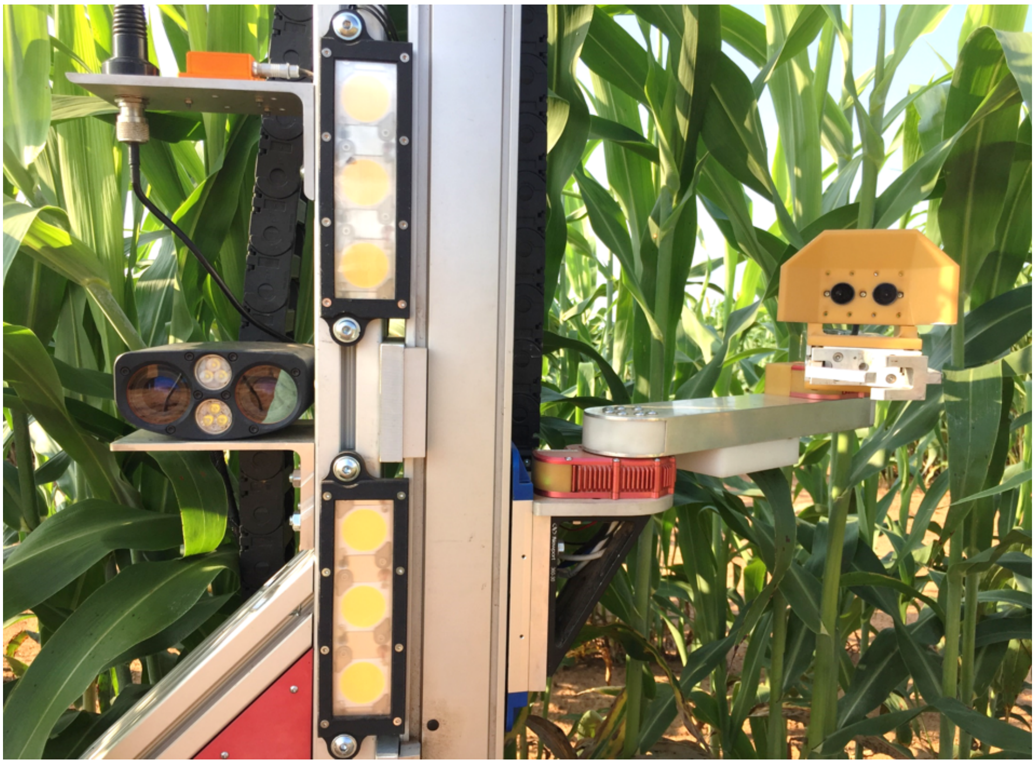


Figure 7.1: Left: The Multisense S7 is mounted horizontally to the robot mast; Center: Two custom flash boards are mounted vertically on the mast between the Multisense S7 and the first joint of the manipulator; Right: The DUO3D camera is still mounted to the manipulator, but it is unused

FPGA), and a focus on pruning the stitched point cloud before performing computationally intensive tasks such as patch normal estimation. Grasp points are selected by incorporating new information, such as color, and using patch normals to aggressively prune the stitched cloud. The following paragraphs describe this algorithm in detail.

As described in 6.1.2, five point clouds are stitched together to increase the visibility of occluded stalks. The incorporation of wheel odometry to initialize cloud locations before stitching reduces computation time significantly compared to the method described in 6.1.2. The time required to stitch clouds drops from an average of 10 seconds without wheel odometry to an average of 3.2 seconds. The incorporation of wheel odometry also reduces the likelihood of the ICP algorithm finding a local minimum. An example stitched point cloud is visible in Figure 7.2a.

The S7 produces point clouds that are approximately twice as large as those described in Chapter 6, so point cloud pruning is especially important to reduce processing time. A simple first step is to prune the regions of the stitched point cloud that are outside of the reach space of the manipulator. A second prunable region consists of colors not associated with plant stalks. While the algorithm described in

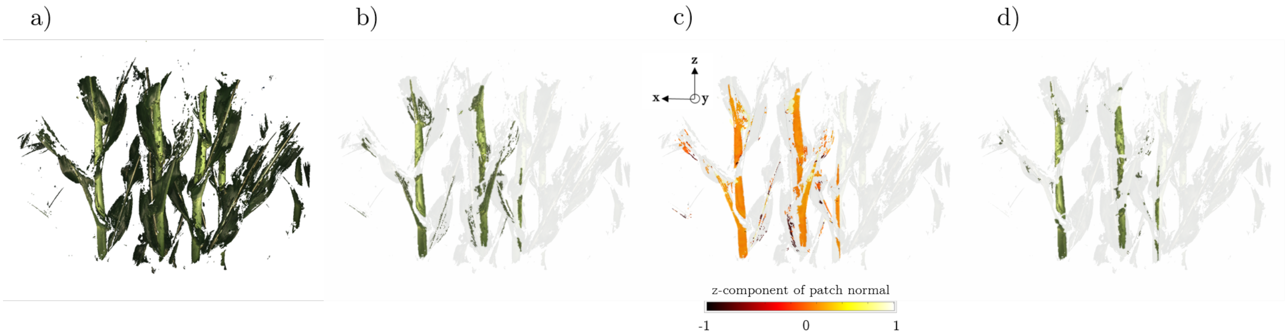


Figure 7.2: a) Stitched point cloud; b) Point cloud after manipulator range thresholding and color pruning; c) Point cloud colorized to indicate the absolute value of the z-component of patch normals; d) Point cloud filtered based on the z-component of patch normals.

6.1.3 intentionally avoids using plant color information, the active flash has proven highly reliable at producing consistent color. An analysis of sorghum and corn images collected from the robot in 2016 and 2017 indicate that stalks are a lighter shade of green than leaves, which presents an opportunity to reduce the size of the point cloud. The combined effect of pruning based on manipulator reach and color reduces the size of point clouds approximately 95%, from 1M points to approximately 40k points. The result of pruning is visible in Figure 7.2b, and the color thresholds are outlined in Table 7.1.

Table 7.1: Grasp Point Detection Variables and Values

Variable	Value	Description
N_{slices}	6	Number of slices
step	0.1m	Height increment of slices
z_{max}	0.2	Maximum threshold on z-component of patch normals
$color_{min}$	(70, 80, 40)	Minimum threshold on RGB value of each point
$r_{neighbor}$	10mm	Radius for nearest neighbor search
r_{normal}	10mm	Radius for normal patch estimation
$hotspot_{factor}$	100	Histogram seed point threshold factor
$noise_{factor}$	200	Histogram noise threshold factor

In Chapter 6.1.3, point cloud normals were averaged in each histogram bin to weight the value of each bin. This was done to reduce the effect of noise in point normal estimation. The point clouds generated by the S7 are of high enough quality that individual point normal estimation is reliable. This is visualized in Figure 7.2c. Individual points that have normal components in the z-axis outside of the threshold outlined in Table 7.1 are pruned from the cloud. Normal estimation is the most computationally intensive component of the algorithm after cloud stitching, computation on the pruned cloud takes only 0.3-0.4 seconds.

As described in 6.1.3, the point cloud is sliced and collapsed onto 2D histograms. However, unlike the previous algorithm, the noise and seed point thresholds are automatically determined by the size of the cloud. The size of the cloud provides an indication of the global cloud density, which drives the two region growing thresholds. The values used to set the seed threshold (β) and noise threshold (α) are calculated according to the following method:

$$\alpha = \frac{\text{cloud}_{\text{size}}}{\text{noise}_{\text{factor}}}$$

$$\beta = \frac{\text{cloud}_{\text{size}}}{\text{hotspot}_{\text{factor}}}$$

The 8-point region growing algorithm implemented for centroid detection is the same as that outlined in 6.1.3.

After centroids are located in the histogram, they are projected to 3D. The 3D centroid locations are then refined using a nearest neighbor search, and the centroid is reassigned to the average location of its neighbors. This counteracts an occasional failure mode where a centroid is projected in between a stalk and the leaf emerging from the stalk. The value assigned to the search radius is specified in Table 7.1.

Nearest neighbor search also accommodates color analysis. When centroids are incorrectly projected to leaves, the average color associated with that centroids neighbors is typically darker than the average color of the total cloud. This generally holds true even after color-based pruning. Thus, if the projected centroid is brighter than the average color of the cloud, it is deemed a stalk location, and the arm planner adds it to a list of locations to grasp.

7.4 Preliminary Results and Discussion

The described algorithm is significantly faster than that outlined in Chapter 6, primarily due to the use of an FPGA for fast correspondence matching and wheel odometry for cloud stitching. In total, the algorithm takes between 3.5 and 4.5 seconds to stitch clouds and detect grasp points. This method of grasp point detection was tested on 30 datasets, each consisting of five image pairs and corresponding wheel odometry. The proposed algorithm correctly identified 210 grasp points with only 3 errors, however more data is necessary to fully quantify reliability. A representative output is displayed in Figure 7.3.

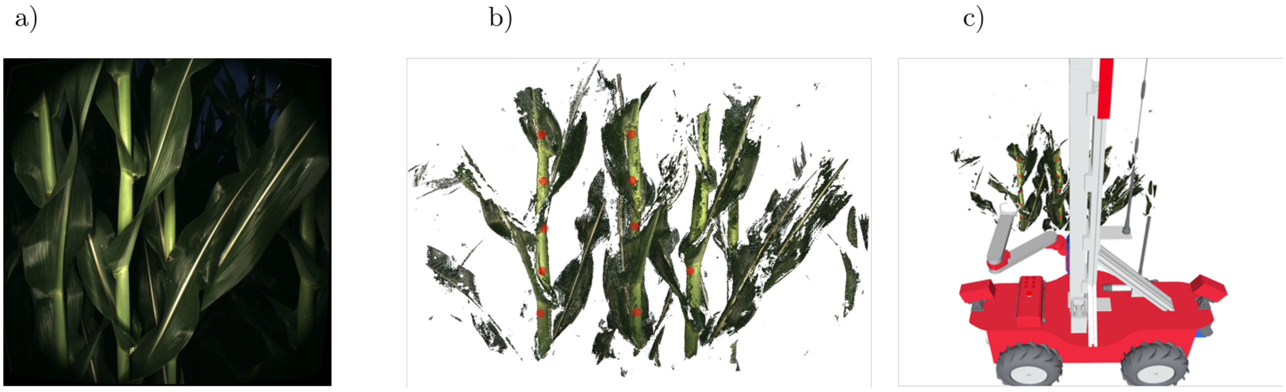


Figure 7.3: a) Rectified color image from the left camera; b) Grasp points visualized as red spheres within a point cloud; c) Point cloud and grasp points visualized relative to the Robotanist.

Corn stalks and sorghum stalks are similar, but the differences are worth noting. Corn leaves exhibit a more vertical orientation than sorghum leaves, which may cause more stalk occlusion but also may be less likely to block the camera lenses. Corn stalks also exhibit a more consistent diameter than that of sorghum, but stalk diameter should not affect algorithm performance. Sorghum leaves are a similar darkness to corn leaves, but the color thresholding parameters have not been extensively tested on sorghum images. Further testing is necessary to conclusively evaluate the algorithm's performance on sorghum, but preliminary results on corn stalks are compelling.

Chapter 8

Conclusion

8.1 Summary

This paper presents the design decisions involved in developing a ground platform, manipulator, and end-effector for autonomous application of a rind penetrometer test to sorghum stalks. Three different stalk detection algorithms are detailed, including necessary hardware, runtime, and performance. The stalk detection algorithms are validated on images of *Sorghum bicolor*, and stalk detection and grasping are demonstrated in situ - a first for automated application of the rind penetrometer.

8.2 Applications

The proof-of-concept outlined in this thesis can be applied to plant phenotyping uses other than rind penetrometer measurements. Near-infrared (NIR), short wave infrared (SWIR) and Raman spectroscopy are common processes deployed in the phenotyping pipeline [53], but they require physical contact or close proximity to the plant organ being measured. The system described in this thesis could be easily adapted to deploy a portable spectrometer against the wall of a stalk.

Some phenotyping tests require a preprocessing step that could be implemented by a robot, such as injecting plant stalks with fungus to measure stalk rot resistance. Autonomous stalk injection would require extra mechanical design, but the manipulator architecture and computer vision algorithms would remain unchanged.

The stalk detection algorithms outlined in this thesis are not limited to sorghum stalks. If fact, the stalks of *Sorghum bicolor* may be more difficult to detect than those of similar crops. This is because of the tendency for multiple sorghum stalks to grow from a single seed (known as tillering) and a lack of breeding focused on crop uniformity. Deploying the rind penetrometer system on corn stalks or sugar cane would require little to no modification.

Applications in the horticultural industry would require more research and development, but are within the realm of possibility. These include detecting stems for autonomous flower harvesting, or detecting peduncles for autonomous harvesting of cucumber, eggplant, or melons. There is significant industry demand for greenhouse automation, especially in Japan where labor costs are particularly high.

8.3 Future Work

Future work in the sorghum phenotyping space can be divided into two areas: applying the described system to other plant organs, and improving the sorghum stalk detection pipeline.

Elements of the computer vision architecture developed for this project can be applied to detecting other plant organs, such as leaves. The stereo camera, synchronized flash, and point cloud stitching procedure are applicable to vision-based detection of any part of a plant, and leaf detection would benefit from detecting stalks and removing them from the image. However, unlike cylindrically-parameterized stalks, leaves require more complex parameterization. A manipulator designed to grasp leaves would require more degrees of freedom and more complex path planning, but developing such a system is entirely within the realm of possibility.

Several methods of detecting stalks are not explored in this thesis. These include the use of 2D stalk detection methods such as neural networks, Hough Transforms, and Frangi filters [22]. Neural networks, especially region-based convolutional neural networks (RCNNs) and fully convolutional network (FCNs) were recently demonstrated to be effective at detecting stalks in 2D images [24] at high rates (2-5 Hz). Preliminary stalk detection results from generative adversarial networks indicate even higher throughput (20 Hz) and remarkably accurate results at the pixel level. Despite the fact that point clouds contain more information than 2D images, state-of-the-art machine learning methods for 2D classification are far superior than 3D machine learning methods. Future work will include performing 2D classification on a single image from a stereo pair, and projecting the classified points to 3D.

Reliable, autonomous phenotyping would be a boon to the agricultural industry and humanity as a whole. Crop efficiency is critical to boosting yields as well as reducing environmental impact. Robotics will play a critical role in improving food security, and the development of stalk detection algorithms is one small step toward a 21st century in which food systems are stable and accessible throughout the world.

Table 8.1: Imaging Sensors and Specifications

Imaging Sensors	Min. Range (mm)	Approx. Cost	Resolution (MP)	Outdoor Capability	Max. Frame Rate (fps)	Shortcoming
PMD Camcube 3.0	300	$25,000	0.04	Yes	40	Cost/Resolution
Microsoft Kinect	800	$200	0.3	No	30	Outdoor Operation
PMD CamBoard Nano	100	$900	0.04	Unclear	45	Resolution
Point Grey Grasshopper3	100	$975	3.2	Yes	121	
DUO3D Stereo Camera	100	$700	0.3	Yes	15	
ZED Stereo Camera	500	$450	5.5	Yes	15	Minimum Range
Bumblebee2 Stereo Camera	500	$2,400	0.8	Yes	20	Minimum Range
Bumblebee XB3 Stereo Camera	500	$3,500	0.8	Yes	20	Minimum Range
Multisense S7	200	$7,000	4	Yes	10	
Velodyne Puck	1000	$8,000	0.3*	Yes	1*	Minimum Range
SICK TiM Series Planar LiDAR	50	$2,000	0.0003	Yes	15	Depth Estimation Error
SICK OD2 triangulation sensor	50	$1,000	0.000001	Yes	2000	Single Beam
Custom Stereo Camera	250	$5,000	9	Yes	5	

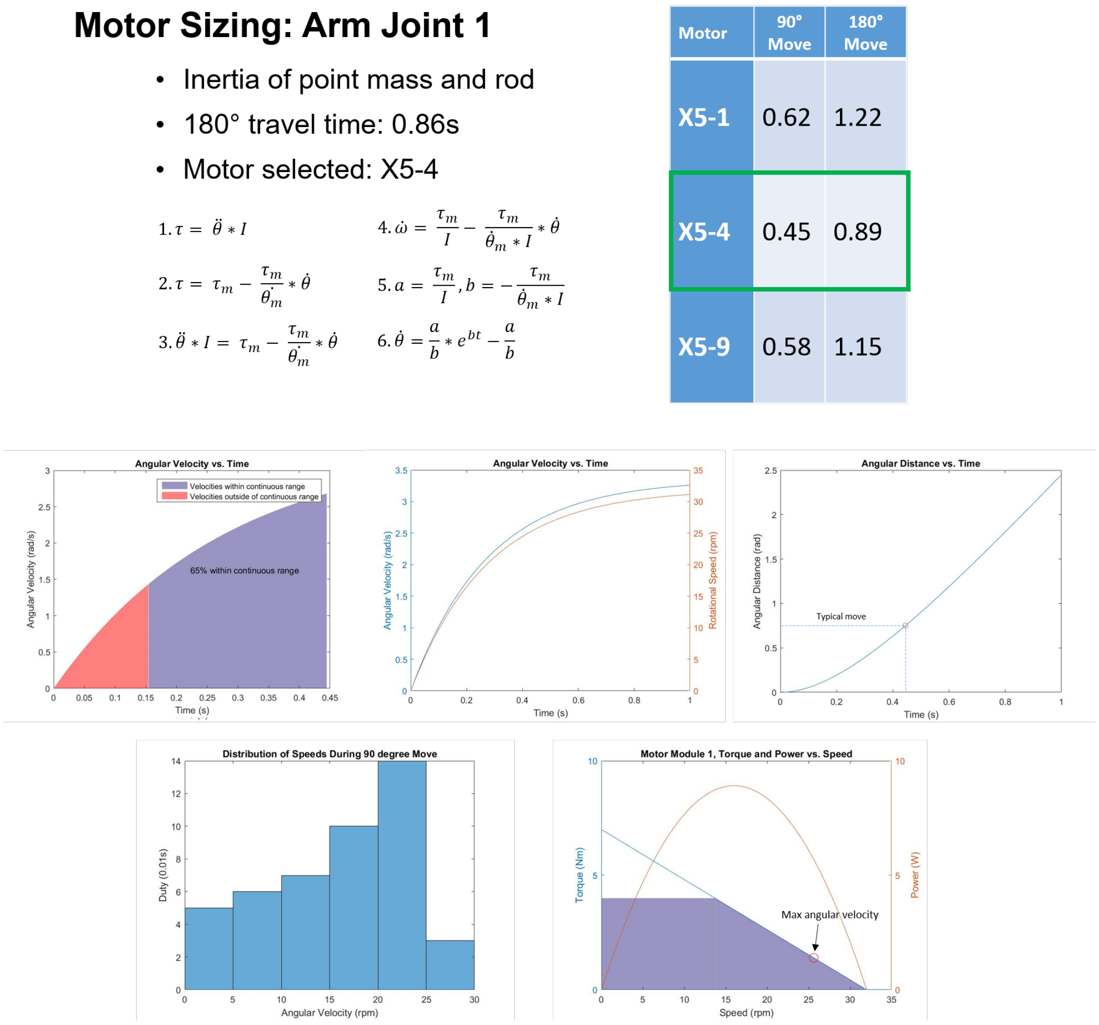
*300,000 pps

Table 8.2: Manipulator Performance Specifications

Companies	**Manipulator**	**DOF**	**Reach (mm)**	**Mass (kg)**	**Payload (kg)**	**Cost**	**IP Class**
ABB	IRB120	6	580	25	3	$30,000	30
Barrett	WAM	4	1000	25	4	$160,000	Unspecified
Kinova	JACO2	4	750	3.6	4.4	$22,950	X2
Kinova	JACO2	6	900	4.4	2.6	$29,950	X2
Kinova	MICO2	4	550	3.8	5.2	$16,900	X2
Kinova	MICO2	6	700	4.6	2.1	$20,900	X2
Robai	Cyton Gamma 300	7	480	1.2	0.3	$5,000	X2
Universal Robots	UR3	6	500	11	6.6	$23,000	54
Universal Robots	UR5	6	850	18.4	5	$35,000	54
Universal Robots	UR10	6	1300	28.9	10	$45,000	54
GearWurx	ARM 3.0	6	2200	9.25	4.5	$5,995	Unspecified
Robotis	Manipulator-H	6	633	5.5	3	$18,900	54
Schunk	LWA-4D	7	1113	18	10	$75,000	54
AUBO	I-5	6	924	24	5	$15,000	54

Appendix A

Manipulator Sizing



Motor Sizing: Arm Joint 1

- Inertia of point mass and rod
- 180° travel time: 0.86s
- Motor selected: X5-4

$1.\ \tau = \ddot{\theta} * I$

$2.\ \tau = \tau_m - \frac{\tau_m}{\dot{\theta}_m} * \dot{\theta}$

$3.\ \ddot{\theta} * I = \tau_m - \frac{\tau_m}{\dot{\theta}_m} * \dot{\theta}$

$4.\ \dot{\omega} = \frac{\tau_m}{I} - \frac{\tau_m}{\dot{\theta}_m * I} * \dot{\theta}$

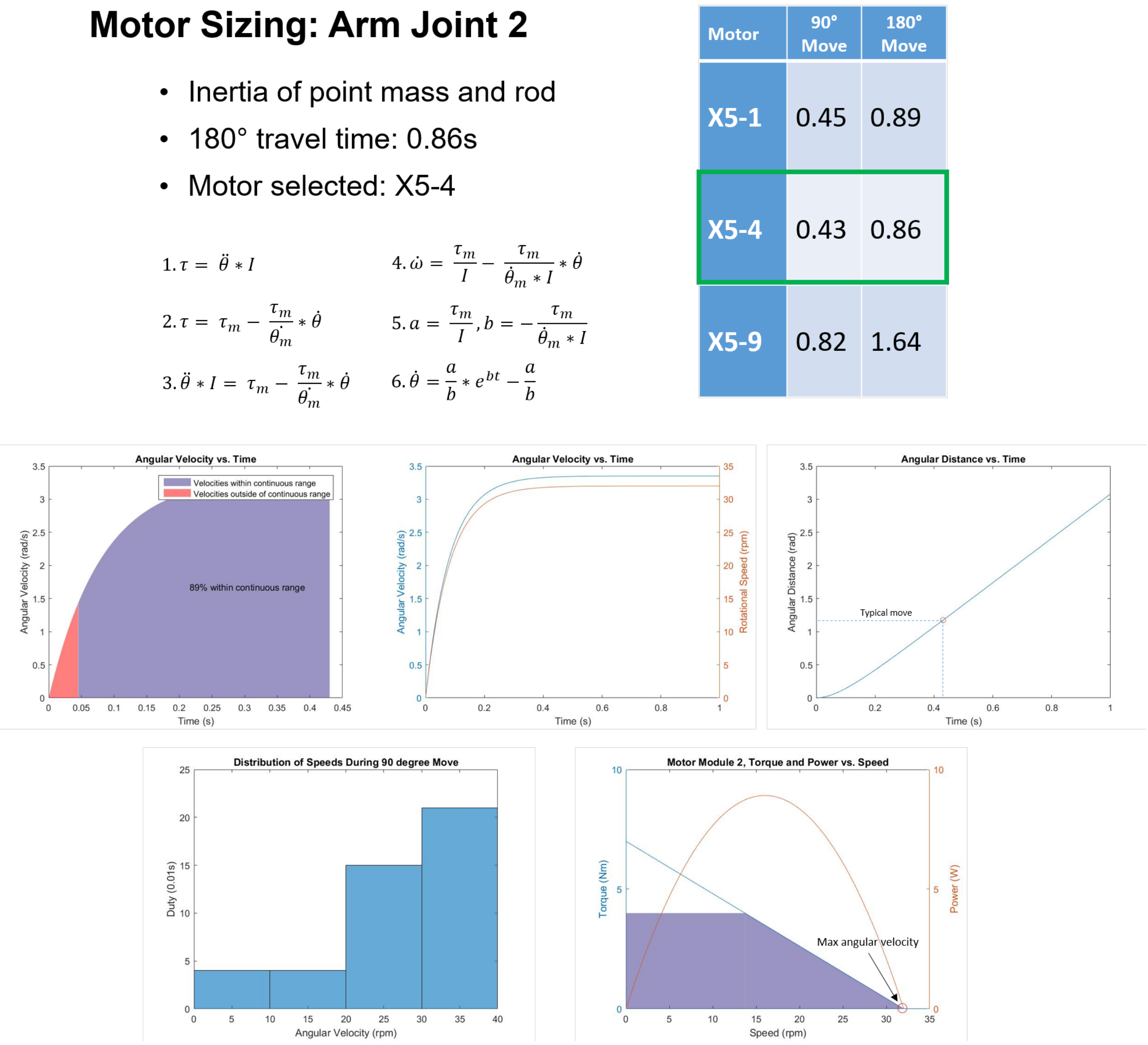
$5.\ a = \frac{\tau_m}{I}, b = -\frac{\tau_m}{\dot{\theta}_m * I}$

$6.\ \dot{\theta} = \frac{a}{b} * e^{bt} - \frac{a}{b}$

Motor	90° Move	180° Move
X5-1	0.62	1.22
X5-4	0.45	0.89
X5-9	0.58	1.15

Angular Velocity vs. Time
Velocities within continuous range
Velocities outside of continuous range
65% within continuous range
Angular Velocity (rad/s)
Time (s)
Angular Velocity vs. Time
Angular Velocity (rad/s)
Rotational Speed (rpm)
Time (s)
Angular Distance vs. Time
Typical move
Angular Distance (rad)
Time (s)
Distribution of Speeds During 90 degree Move
Duty (0.01s)
Angular Velocity (rpm)
Motor Module 1, Torque and Power vs. Speed
Torque (Nm)
Power (W)
Max angular velocity
Speed (rpm)

Figure A.1: Motor sizing calculations for the first rotary joint consist of estimating rotational speed of the arm and heat dissipation in the motor.



Motor Sizing: Arm Joint 2

- Inertia of point mass and rod
- 180° travel time: 0.86s
- Motor selected: X5-4

$1.\ \tau = \ddot{\theta} * I$

$2.\ \tau = \tau_m - \frac{\tau_m}{\dot{\theta}_m} * \dot{\theta}$

$3.\ \ddot{\theta} * I = \tau_m - \frac{\tau_m}{\dot{\theta}_m} * \dot{\theta}$

$4.\ \dot{\omega} = \frac{\tau_m}{I} - \frac{\tau_m}{\dot{\theta}_m * I} * \dot{\theta}$

$5.\ a = \frac{\tau_m}{I}, b = -\frac{\tau_m}{\dot{\theta}_m * I}$

$6.\ \dot{\theta} = \frac{a}{b} * e^{bt} - \frac{a}{b}$

Motor	90° Move	180° Move
X5-1	0.45	0.89
X5-4	0.43	0.86
X5-9	0.82	1.64

Figure A.2: Motor sizing calculations for the second rotary joint consist of estimating rotational speed of the arm and heat dissipation in the motor.

Appendix B

Bearing Life Calculations

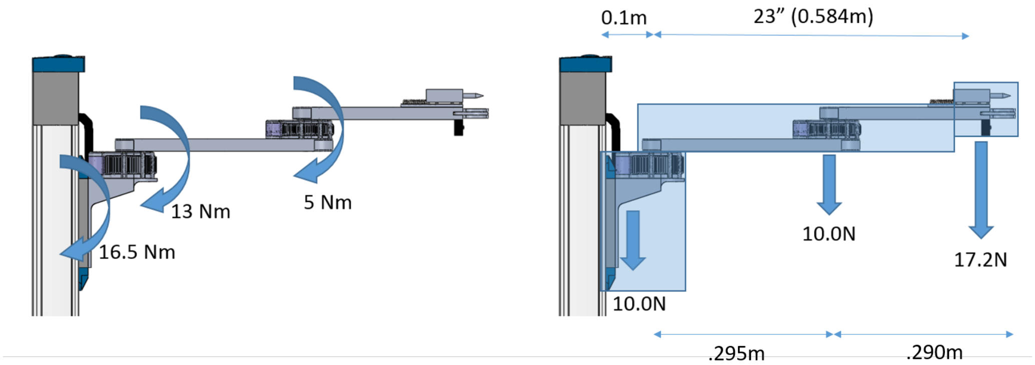
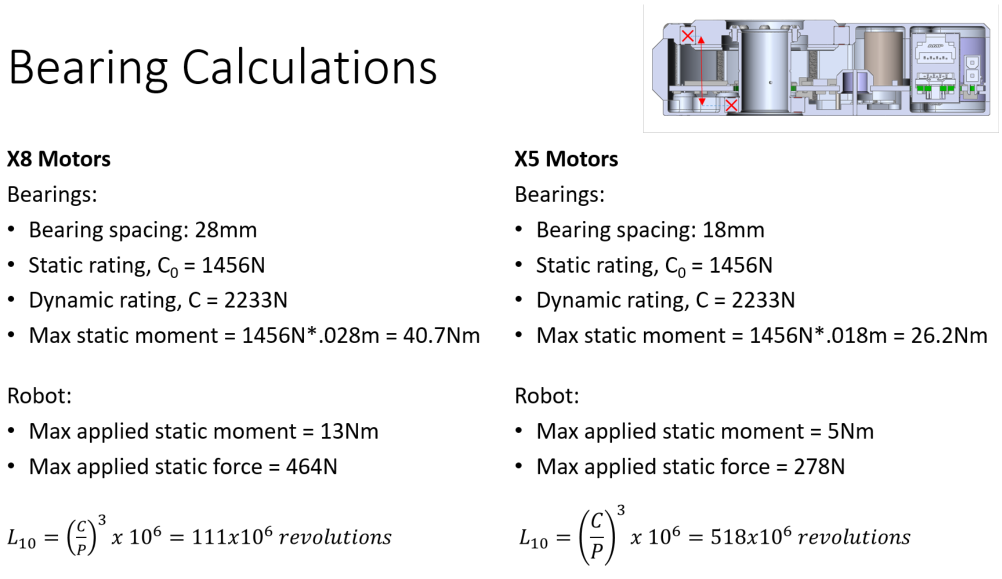


Figure B.1: Preliminary calculations of bending moments on a PRR manipulator.



Bearing Calculations

X8 Motors

Bearings:

- Bearing spacing: 28mm
- Static rating, C_0 = 1456N
- Dynamic rating, C = 2233N
- Max static moment = 1456N*.028m = 40.7Nm

Robot:

- Max applied static moment = 13Nm
- Max applied static force = 464N

$$L_{10} = \left(\frac{C}{P}\right)^3 x\ 10^6 = 111x10^6\ revolutions$$

X5 Motors

Bearings:

- Bearing spacing: 18mm
- Static rating, C_0 = 1456N
- Dynamic rating, C = 2233N
- Max static moment = 1456N*.018m = 26.2Nm

Robot:

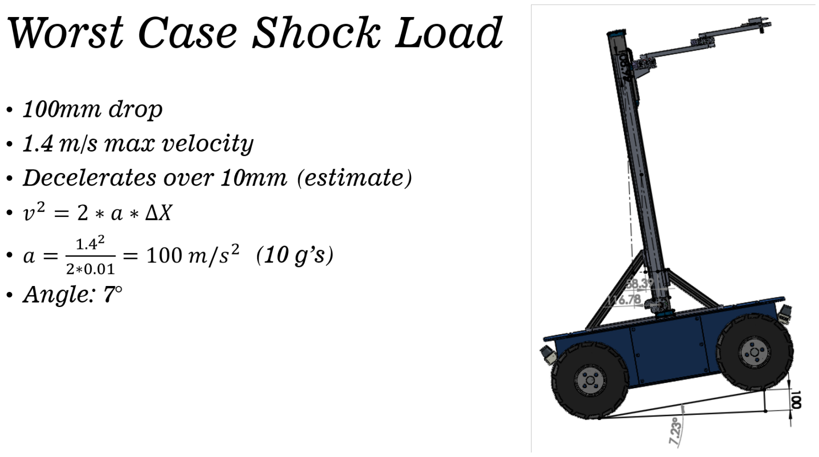
- Max applied static moment = 5Nm
- Max applied static force = 278N

$$L_{10} = \left(\frac{C}{P}\right)^3 x\ 10^6 = 518x10^6\ revolutions$$

Figure B.2: Preliminary bearing life calculations on HEBI X5 motors.

Appendix C

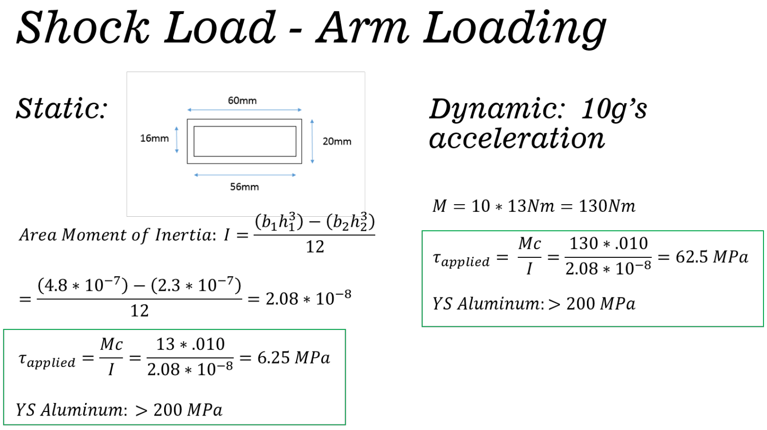
Shock Loading



Worst Case Shock Load

- *100mm drop*
- *1.4 m/s max velocity*
- *Decelerates over 10mm (estimate)*
- $v^2 = 2 * a * \Delta X$
- $a = \frac{1.4^2}{2*0.01} = 100\ m/s^2$ *(10 g's)*
- *Angle: 7°*

Figure C.1: Preliminary calculations of the maximum shock load the robot could experience.



Shock Load - Arm Loading

Static:

60mm
16mm
20mm
56mm

$$Area\ Moment\ of\ Inertia:\ I = \frac{(b_1 h_1^3) - (b_2 h_2^3)}{12}$$

$$= \frac{(4.8 * 10^{-7}) - (2.3 * 10^{-7})}{12} = 2.08 * 10^{-8}$$

$$\tau_{applied} = \frac{Mc}{I} = \frac{13 * .010}{2.08 * 10^{-8}} = 6.25\ MPa$$

$YS\ Aluminum:\ > 200\ MPa$

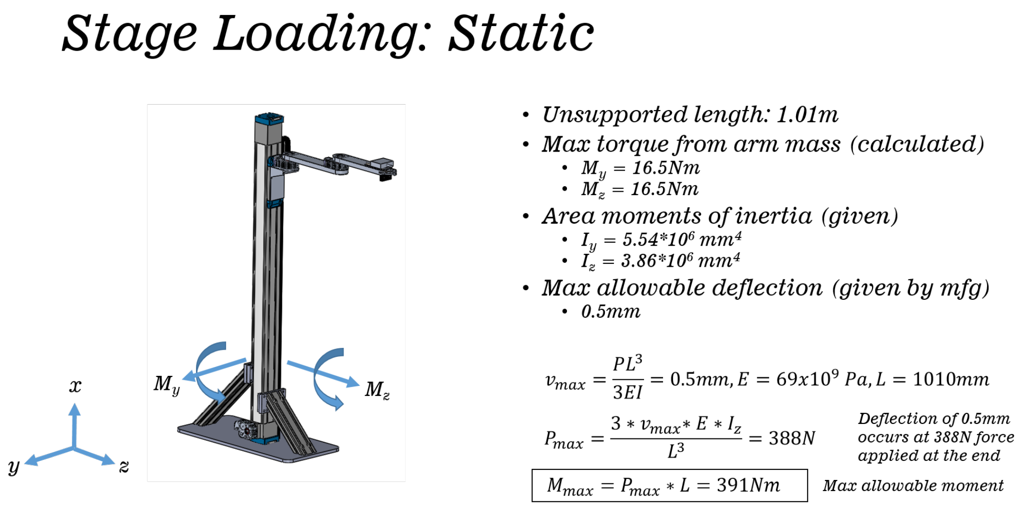
Dynamic: 10g's acceleration

$$M = 10 * 13Nm = 130Nm$$

$$\tau_{applied} = \frac{Mc}{I} = \frac{130 * .010}{2.08 * 10^{-8}} = 62.5\ MPa$$

$YS\ Aluminum: > 200\ MPa$

Figure C.2: Preliminary calculations of the maximum stress applied to the aluminum arms under a 10g shock load.



Stage Loading: Static

- *Unsupported length: 1.01m*
- *Max torque from arm mass (calculated)*
  - $M_y = 16.5Nm$
  - $M_z = 16.5Nm$
- *Area moments of inertia (given)*
  - $I_y = 5.54*10^6\ mm^4$
  - $I_z = 3.86*10^6\ mm^4$
- *Max allowable deflection (given by mfg)*
  - *0.5mm*

$$v_{max} = \frac{PL^3}{3EI} = 0.5mm, E = 69x10^9\ Pa, L = 1010mm$$

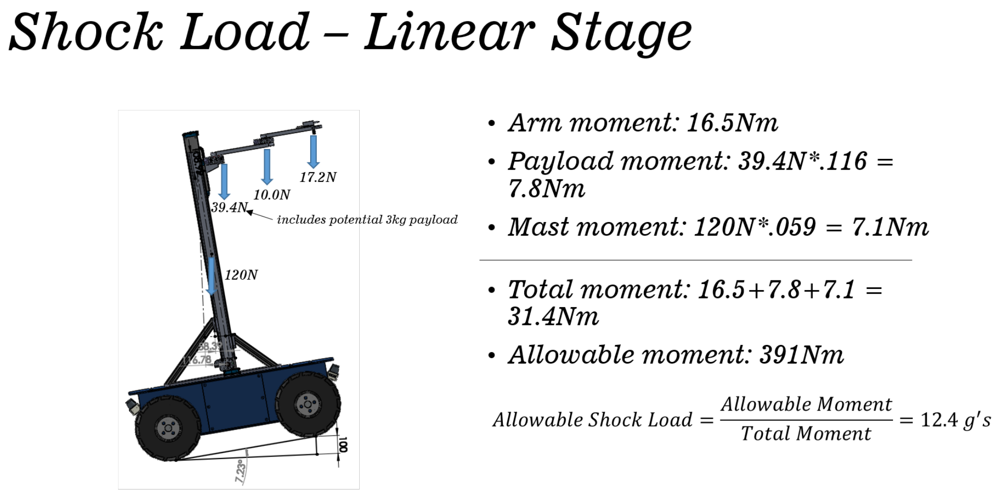
$$P_{max} = \frac{3 * v_{max} * E * I_z}{L^3} = 388N$$

Deflection of 0.5mm occurs at 388N force applied at the end

$$\boxed{M_{max} = P_{max} * L = 391Nm}$$

Max allowable moment

Figure C.3: Preliminary calculations analyzing the moment that can be applied to the linear stage before exceeding the manufacturer’s suggested maximum deflection of 0.5mm.



Shock Load – Linear Stage

- *Arm moment: 16.5Nm*
- *Payload moment: 39.4N*.116 = 7.8Nm*
- *Mast moment: 120N*.059 = 7.1Nm*

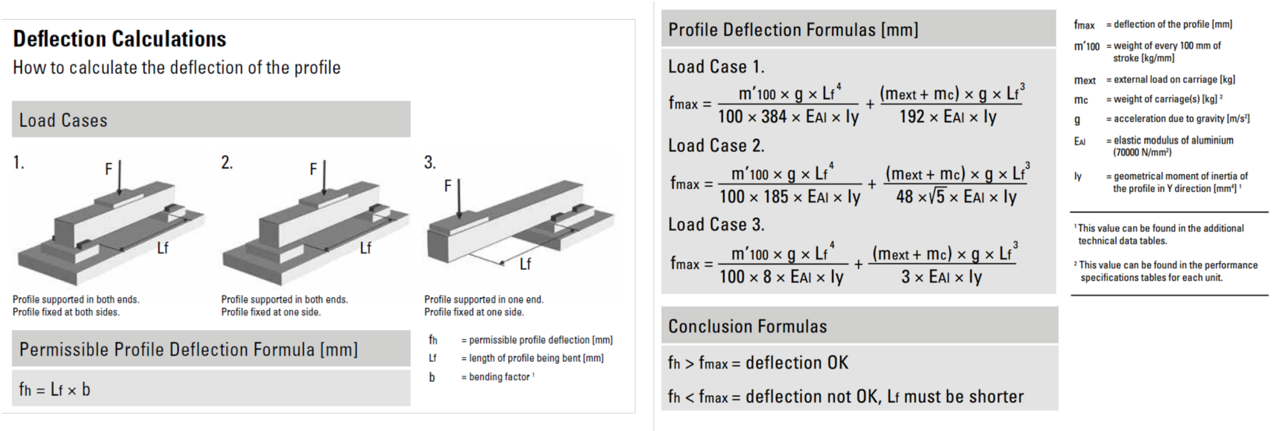
- *Total moment: 16.5+7.8+7.1 = 31.4Nm*
- *Allowable moment: 391Nm*

$$Allowable\ Shock\ Load = \frac{Allowable\ Moment}{Total\ Moment} = 12.4\ g's$$

Figure C.4: Preliminary calculations analyzing the allowable shock load on the linear stage given the manufacturer’s suggested maximum deflection of 0.5mm.

Appendix D

Linear Stage Specifications



1.
F
Lf
2.
F
Lf
3.
F
Lf

Deflection Calculations

How to calculate the deflection of the profile

Load Cases

1. Profile supported in both ends. Profile fixed at both sides.

2. Profile supported in both ends. Profile fixed at one side.

3. Profile supported in one end. Profile fixed at one side.

Permissible Profile Deflection Formula [mm]

$$f_h = L_f \times b$$

f_h = permissible profile deflection [mm]

L_f = length of profile being bent [mm]

b = bending factor [1]

Profile Deflection Formulas [mm]

Load Case 1.

$$f_{max} = \frac{m'_{100} \times g \times L_f^4}{100 \times 384 \times E_{Al} \times I_y} + \frac{(m_{ext} + m_c) \times g \times L_f^3}{192 \times E_{Al} \times I_y}$$

Load Case 2.

$$f_{max} = \frac{m'_{100} \times g \times L_f^4}{100 \times 185 \times E_{Al} \times I_y} + \frac{(m_{ext} + m_c) \times g \times L_f^3}{48 \times \sqrt{5} \times E_{Al} \times I_y}$$

Load Case 3.

$$f_{max} = \frac{m'_{100} \times g \times L_f^4}{100 \times 8 \times E_{Al} \times I_y} + \frac{(m_{ext} + m_c) \times g \times L_f^3}{3 \times E_{Al} \times I_y}$$

f_{max} = deflection of the profile [mm]

m'_{100} = weight of every 100 mm of stroke [kg/mm]

m_{ext} = external load on carriage [kg]

m_c = weight of carriage(s) [kg] [2]

g = acceleration due to gravity [m/s^2]

E_{Al} = elastic modulus of aluminium (70000 N/mm^2)

I_y = geometrical moment of inertia of the profile in Y direction [mm^4] [1]

[1] This value can be found in the additional technical data tables.

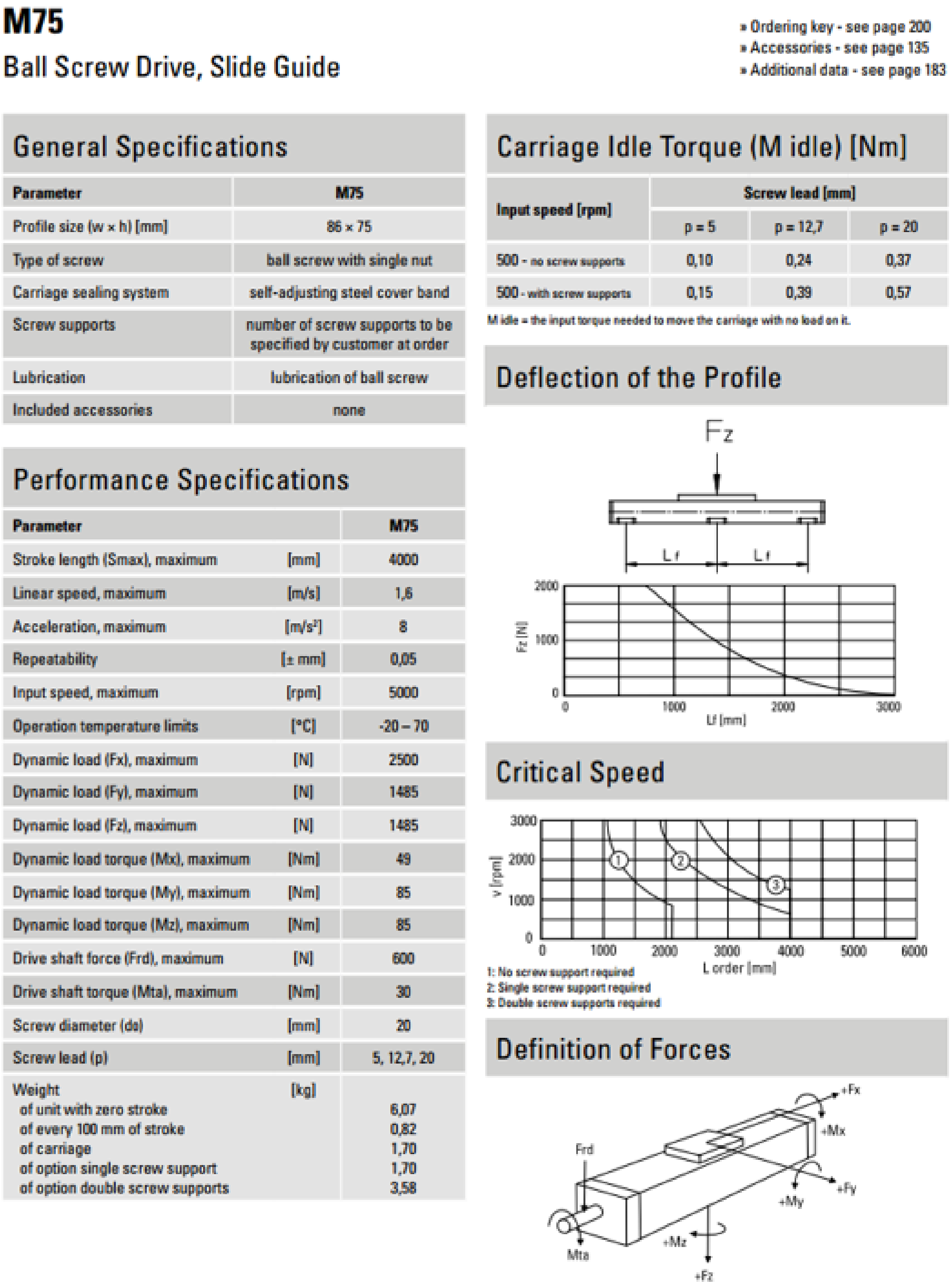
[2] This value can be found in the performance specifications tables for each unit.

Conclusion Formulas

$f_h > f_{max}$ = deflection OK

$f_h < f_{max}$ = deflection not OK, L_f must be shorter

Figure D.1: Manufacturer's suggested method of calculating deflection based on different loading configurations.



M75

Ball Screw Drive, Slide Guide

» Ordering key - see page 200
» Accessories - see page 135
» Additional data - see page 183

General Specifications

Parameter	M75
Profile size (w × h) [mm]	86 × 75
Type of screw	ball screw with single nut
Carriage sealing system	self-adjusting steel cover band
Screw supports	number of screw supports to be specified by customer at order
Lubrication	lubrication of ball screw
Included accessories	none

Performance Specifications

Parameter		M75
Stroke length (Smax), maximum	[mm]	4000
Linear speed, maximum	[m/s]	1,6
Acceleration, maximum	[m/s²]	8
Repeatability	[± mm]	0,05
Input speed, maximum	[rpm]	5000
Operation temperature limits	[°C]	-20 – 70
Dynamic load (Fx), maximum	[N]	2500
Dynamic load (Fy), maximum	[N]	1485
Dynamic load (Fz), maximum	[N]	1485
Dynamic load torque (Mx), maximum	[Nm]	49
Dynamic load torque (My), maximum	[Nm]	85
Dynamic load torque (Mz), maximum	[Nm]	85
Drive shaft force (Frd), maximum	[N]	600
Drive shaft torque (Mta), maximum	[Nm]	30
Screw diameter (d0)	[mm]	20
Screw lead (p)	[mm]	5, 12,7, 20
Weight of unit with zero stroke of every 100 mm of stroke of carriage of option single screw support of option double screw supports	[kg]	 6,07 0,82 1,70 1,70 3,58

Carriage Idle Torque (M idle) [Nm]

Input speed [rpm]	Screw lead [mm]		
	p = 5	p = 12,7	p = 20
500 - no screw supports	0,10	0,24	0,37
500 - with screw supports	0,15	0,39	0,57

M idle = the input torque needed to move the carriage with no load on it.

Deflection of the Profile

Critical Speed

1: No screw support required
2: Single screw support required
3: Double screw supports required

Definition of Forces

Figure D.2: Manufacturer's specifications regarding maximum loading scenarios on the carriage.

Appendix E

Two One-sided Equivalency Test

- Null Hypothesis #1: The population means differ more than 15% from the average force reading
- Null Hypothesis #2: The population means differ more than -15% from the average force reading

Average Force Reading: 17.15 lbs

15% of Average Force Reading: 2.57 lbs

Set the equivalence threshold at ±2.57lbs

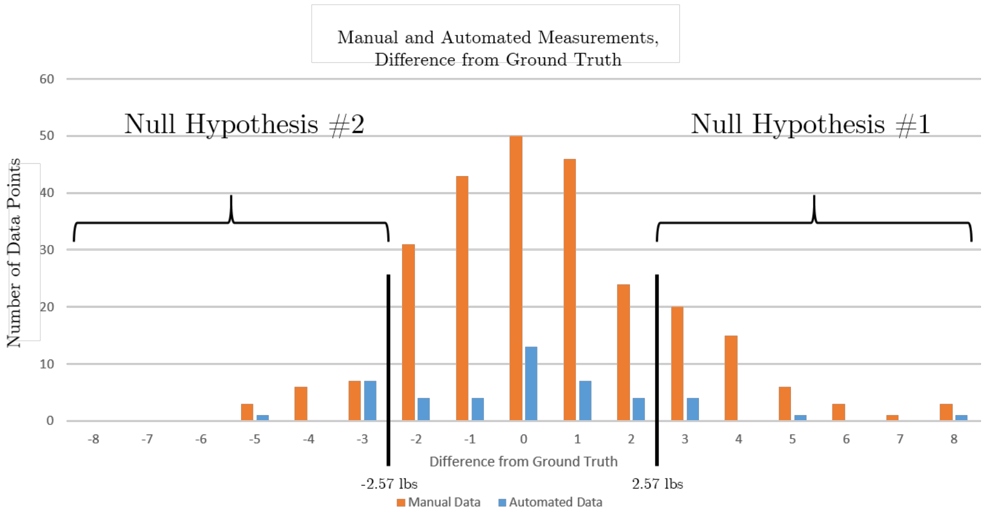
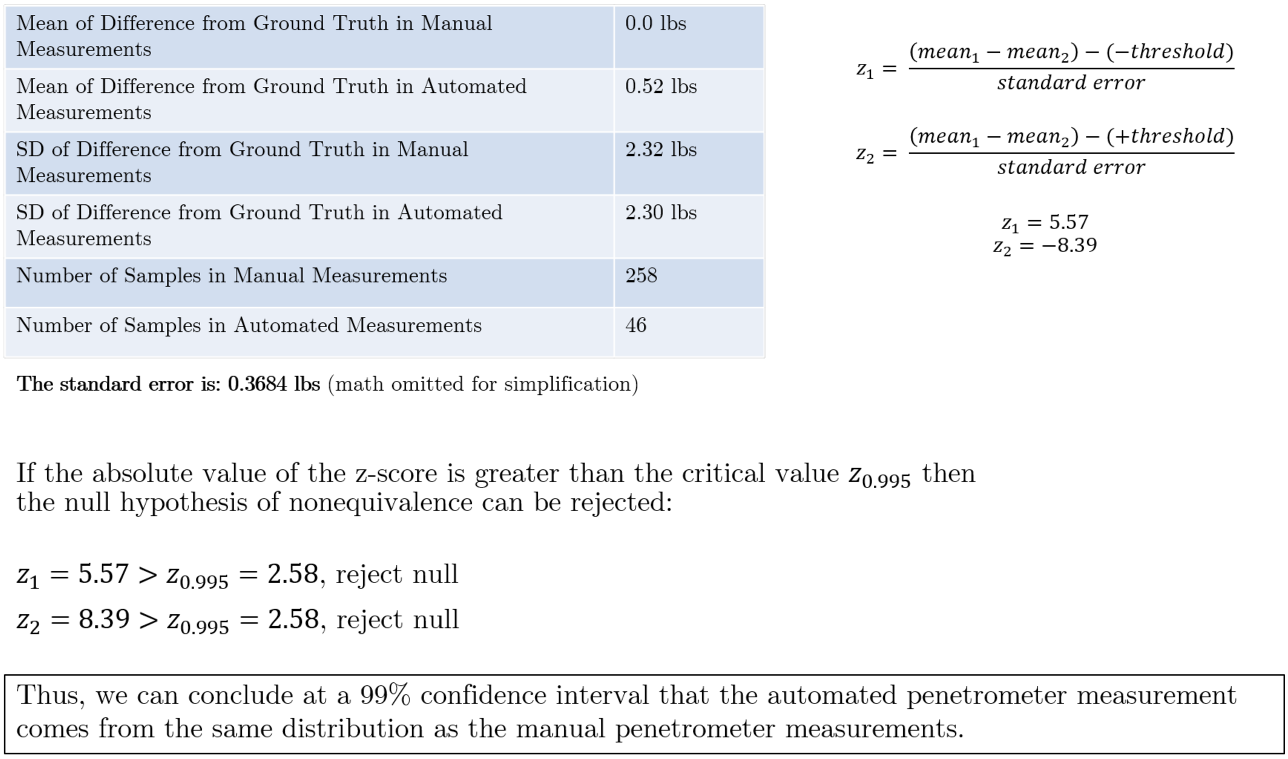


Figure E.1: The two null hypotheses assume that the population means differ by more than ±, and these are depicted in the histogram.



Mean of Difference from Ground Truth in Manual Measurements	0.0 lbs
Mean of Difference from Ground Truth in Automated Measurements	0.52 lbs
SD of Difference from Ground Truth in Manual Measurements	2.32 lbs
SD of Difference from Ground Truth in Automated Measurements	2.30 lbs
Number of Samples in Manual Measurements	258
Number of Samples in Automated Measurements	46

The standard error is: 0.3684 lbs (math omitted for simplification)

$$z_1 = \frac{(mean_1 - mean_2) - (-threshold)}{standard\ error}$$

$$z_2 = \frac{(mean_1 - mean_2) - (+threshold)}{standard\ error}$$

$$z_1 = 5.57$$
$$z_2 = -8.39$$

If the absolute value of the z-score is greater than the critical value $z_{0.995}$ then the null hypothesis of nonequivalence can be rejected:

$z_1 = 5.57 > z_{0.995} = 2.58$, reject null

$z_2 = 8.39 > z_{0.995} = 2.58$, reject null

Thus, we can conclude at a 99% confidence interval that the automated penetrometer measurement comes from the same distribution as the manual penetrometer measurements.

Figure E.2: The z-scores calculated from the two one-sided tests are greater than the critical value, indicating that the null hypotheses can be rejected.

References

[1] N. Alexandratos and J. Bruinsma, “World agriculture towards 2030/2050: The 2012 revision,” *Food and Agriculture Organization of the United Nations*, 2012.

[2] J. Mullet *et al*, “Energy sorghum – a genetic model for the design of C4 grass bioenergy crops,” *Journal of Experimental Botany*, 2014.

[3] B. Metz, O. Davidson, P. Bosch, R. Dave, and L. Meyer, *Climate Change 2007: Mitigation. Contribution of working group III to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge, United Kingdom and New York, NY, USA: Cambridge University Press, 2007, p. 499.

[4] A. Dhanapal and M. Govindaraj, “Unlimited thirst for genome sequencing, data interpretation, and database usage in genomic era: The road towards fast-track crop plant improvement,” *Genetics Research International*, no. 684321, 2015.

[5] R. T. Furbank and M. Tester, “Phenomics technologies to relieve the phenotyping bottleneck,” *Trends in Plant Science*, vol. 16, no. 12, pp. 635–644, 2011.

[6] C. Ainsworth, “Agriculture: A new breed of edits,” *Nature*, vol. 528, no. 7580, 2015.

[7] E. Waltz, “Crispr-edited crops free to enter market, skip regulation,” *Nature Biotechnology*, vol. 34, no. 582, 2016.

[8] M. Eberius, “Tomato phenotyping: High content screening,” White Paper, LemnaTec GmbH, 2014.

[9] D. Rosenow and J. Dahlberg, *Sorghum: Origin, history, technology, and production.* New York: John Wiley & Sons, 2000.

[10] G. Muscato, M. Prestifilippo, N. Abbate, and I. Rizzuto, "A prototype of an orange picking robot: past history, the new robot and experimental results," *Industrial Robot: An International Journal*, vol. 32, no. 2, pp. 128–138, 2005.

[11] D. Dey, L. Mummert, and R. Sukthankar, "Classification of plant structures from uncalibrated image sequence," in *Proceedings of the 2012 IEEE Workshop on the Appl. of Computer Vision*, Washington D.C., USA, Jan 2016, pp. 329–336.

[12] J. Noordam, J. Hemming, C. van Heerde, F. Golbach, R. van Soest, and E. Wekking, "Automated rose cutting in greenhouses with 3d vision and robotics: Analysis of 3d vision techniques for stem detection," in *International Conference on Sustainable Greenhouse Systems - Greensys2004*, Leuven, Belgium, Oct 31 2005.

[13] S. Xu, Y. Xun, T. Jia, and Q. Yang, "Detection method for the buds on winter vines based on computer vision," in *Seventh International Symposium on Computational Intelligence and Design*, Hangzhou, China, Dec 13-14 2014.

[14] P. Andrade-Sanchez, M. A. Gore, J. T. Heun, K. R. Thorp, A. E. Carmo-Silva, A. N. French, M. E. Salvucci, and J. W. White, "Development and evaluation of a field-based high-throughput phenotyping platform," *Functional Plant Biology*, vol. 41, no. 1, pp. 68–79, 2014.

[15] S. Paulus, J. Dupuis, A.-K. Mahlein, and H. Kuhlmann, "Surface feature based classification of plant organs from 3d laserscanned point clouds for plant phenotyping," *BMC Bioinformatics*, vol. 14, pp. 238–250, 2013.

[16] S. Chaivivatrakul, L. Tang, M. Dailey, and A. Nakarmic, "Automatic morphological trait characterization for corn plants via 3d holographic reconstruction," *Computers and Electronics in Agriculture*, vol. 109, pp. 109–123, 2014.

[17] S. Paulus, J. Dupuis, A.-K. Mahlein, and H. Kuhlmann, "Surface feature based classification of plant organs from 3d laserscanned point clouds for plant phenotyping," *BMC Bioinformatics*, no. 14, pp. 238–250, 2013.

[18] A. Payne, K. Walsh, P. Subedi, and D. Jarvis, "Estimation of mango crop yield using image analysis segmentation method," *Computers and Electronics in Agriculture*, vol. 91, pp. 57–64, 2013.

[19] A. Silwal, J. R. Davidson, M. Karkee, C. Mo, Q. Zhang, and K. Lewis, "Design, integration, and field evaluation of a robotic apple harvester," *Journal of Field Robotics*, vol. 34, no. 4, pp. 1–20, 2016.

[20] S. Nuske, K. Wilshusen, S. Achar, L. Yoder, S. Narasimhan, and S. Singh, "Automated visual yield estimation in vineyards," *Journal of Field Robotics*, vol. 31, no. 5, pp. 837–860, 2014.

[21] S. W. Chen, S. S. Shivakumar, S. Dcunha, J. Das, E. Okon, C. Qu, C. J. Taylor, and V. Kumar, "Counting apples and oranges with deep learning: A data-driven approach," *IEEE Robotics and Automation Letters*, vol. 2, pp. 781–788, 2017.

[22] Z. M. Amean, T. Low, C. McCarthy, and N. Hancock, "Automatic plant branch segmentation and classification using vesselness measure," in *Proceedings of Australasian Conference on Robotics and Automation*, Sydney, Australia, Dec 2-4 2014.

[23] I. Sa, C. Lehnert, A. English, C. McCool, F. Dayoub, B. Upcroft, and T. Perez, "Peduncle detection of sweet pepper for autonomous crop harvesting–combined color and 3-d information," *IEEE ROBOTICS AND AUTOMATION LETTERS*, vol. 2, no. 2, pp. 4323–4344, April 2017.

[24] H. Baweja, T. Parhar, O. Mirbod, and S. Nuske, "A deep learning pipeline for high-throughput measurement of plant stalk count and stalk width," in *11th Conference on Field and Service Robotics*, Zurich, Switzerland, September 2017.

[25] D. Kelbe, J. van Aardt, P. Romanczyk, M. van Leeuwen, and K. Cawse-Nicholson, "Single-scan stem reconstruction using low-resolution terrestrial laser scanner data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 7, pp. 3414–3427, July 2015.

[26] X. Liang, P. Litkey, J. Hyyppa, H. Kaartinen, M. Vastaranta, and M. Holopainen, "Automatic stem mapping using single-scan terrestrial laser scanning," *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING*, vol. 50, no. 2, pp. 661–670, 2012.

[27] D. Wang, M. Hollaus, E. Puttonen, and N. Pfeifer, "Fast and robust stem reconstruction in complex environments using terrestrial laser scanning," in *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLI-B3, Prague, Czech Republic, July 2016, pp. 411–417.

[28] K. Olofsson, J. Holmgren, and H. Olsson, "Tree stem and height measurements using terrestrial laser scanning and the RANSAC algorithm," *Remote Sensing*, vol. 6, pp. 4323–4344, 2014.

[29] G. Kirly and G. Brolly, "Modelling single trees from terrestrial laser scanning data in a forest reserve," *The Photogrammetric Journal of Finland*, vol. 21, no. 1, pp. 37–50, 2008.

[30] X. Liang, V. Kankare, X. Yu, J. Hyyppa, and M. Holopainen, "Automated stem curve measurement using terrestrial laser scanning," *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING*, vol. 52, no. 3, pp. 1739–1748, 2014.

[31] "Mechatronics And Manufacturing Automation," http://nptel.ac.in/courses/112103174/module7/lec5/3.html 2013, [Online; accessed 14-June-2017].

[32] C. B. *et al*, "Harvesting robots for high-value crops: State-of-the-art review and challenges ahead," *Journal of Field Robotics*, vol. 31, no. 6, pp. 888–911, November 2014.

[33] A. Scarfe, "Development of an autonomous kiwifruit harvester," Ph.D. dissertation, Massey University, Manawatu, New Zealand, 2012.

[34] Z. De-An, L. Jidong, J. Wei, Z. Ying, and C. Yu, "Design and control of an apple harvesting robot," *Biosystems Engineering*, vol. 110, pp. 112–122, 2011.

[35] S. Mehta and T. Burks, "Vision-based control of robotic manipulator for citrus harvesting," *Computers and Electronics in Agriculture*, vol. 102, p. 146158, 2014.

[36] N. Irie, N. Taguchi, T. Horie, and T. Ishimatsu, "Asparagus harvesting robot coordinated with 3-d vision sensor," in *IEEE International Conference on Industrial Technology*, Gippsland, Australia, Feb 10 2009.

[37] C. Bac, "Improving obstacle awareness for robotic harvesting of sweet pepper," Ph.D. dissertation, Graduate School of Product Ecology & Resource Conservation (PE&RC), 2015.

[38] E. V. Henten, J. Hemming, B. van Tuijl, J. Kornet, J. Meuleman, J. Bontsema, and E. van Os, "An autonomous robot for harvesting cucumbers in greenhouses," *Autonomous Robots*, vol. 13, p. 241258, 2002.

[39] S. Hayashi, K. Ganno, Y. Ishii, and I. Tanaka, "Robotic harvesting system for eggplants," *Japan Agricultural Research Quarterly*, vol. 36, no. 3, pp. 163–168, 2002.

[40] J. Baeten, K. Donne, S. Boedrij, W. Beckers, and E. Claesen, *Autonomous Fruit Picking Machine: A Robotic Apple Harvester*. Berlin, Heidelberg: Springer, 2008, pp. 531–539.

[41] S. Hayashi, K. Shigematsu, S. Yamamotoa, K. Kobayashia, Y. Kohnob, J. Kamatac, and M. Kuritac, "Evaluation of a strawberry-harvesting robot in a field test," *Biosystems Engineering*, vol. 105, no. 2, pp. 160–171, 2010.

[42] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.

[43] R. Barth, J. Hemming, and E. J. Henten, "Design of an eye-in-hand sensing and servo control framework for harvesting robotics in dense vegetation," *Biosystems Engineering*, no. 146, pp. 71–84, 2016.

[44] S. Jian, "Research on image-based fuzzy visual servo for picking robot," in *COMPUTER AND COMPUTING TECHNOLOGIES IN AGRICULTURE II, VOLUME 1*, Beijing, China, Oct 18-20 2008, pp. 751–761.

[45] T. Mueller-Sim, M. Jenkins, J. Abel, and G. Kantor, "The Robotanist: A Ground-Based Agricultural Robot for High-Throughput Crop Phenotyping," in *IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, Singapore, May 29-June 3 2017.

[46] R. Coulter, "Implementation of the pure pursuit path tracking algorithm," Carnegie Mellon University - Robotics Institute, Tech. Rep., January 1992.

[47] R. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[48] J. Peiffer, S. A. Flint-Garcia, N. D. Leon, M. D. McMullen, S. M. Kaeppler, and E. S. Buckler, “The genetic architecture of maize stalk strength,” *PLOS One*, vol. 8, no. 6, June 2013.

[49] N. Sclater, *Mechanisms and Mechanical Devices Sourcebook.* New York, NY: McGraw-Hill, 2007, p. 55.

[50] M. Fischler and R. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Graphics and Image Processing*, vol. 24, no. 6, March 1980.

[51] Z. Pothen and S. Nuske, “Texture-based fruit detection via images using the smooth patterns on the fruit,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2016.

[52] R. B. Rusu and S. Cousins, “3D is here: Point Cloud Library (PCL),” in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[53] H. Schulz and M. Baranska, “Identification and quantification of valuable plant substances by ir and raman spectroscopy,” *Graphics and Image Processing*, vol. 47, pp. 13–25, June 2007.